VIC rapport

ÅRGÅNG 3 NR 2
PRIS 15:- inkl moms



Forth – ett annorlunda språk
Skäftingebackens datorörnar
Electronic Arts
Äventyr på VIC 20/64

473–2

Tips och förslag



VIC rapport startar en bank i vilken vi samlar tips och förslag av alla sorter till din **VIC 20** och **VIC 64.** De kan innefatta allt ifrån sladdanslutning till maskinkods-programmering.

VIC rapport kommer att publicera dessa tips och förslag löpande.

Skriv ner dina tips och förslag och skicka dem till

VIC rapport, Tips och förslag
Box 420 54
126 12 Stockholm

Allt som publiceras belönas!

Ledaren

Vi har med glädje konstaterat att antalet VIC-klubbar har ökat runt om i landet. Traditionellt har VIC-klubbarna startats av en eller flera VIC-entusiaster utan något annat gemensamt än sitt stora intresse för VIC-datorer.

Idag har det dykt upp en ny form av VIC-klubbar. Det är personalgrupper på företag som insett vikten av datakunskaper. De har startat klubbar som håller till i företagets lokaler på kvällstid.

En annan typ av företagsbundna VIC-klubbar är storföretag som centralt ordnar kurser för företagets anställda för att de ska kunna hålla sig á jour med utvecklingen på datasidan. På en del företag lånar man t o m ut datorer till personalen.

Oavsett former och hur VIC-klubbarna startas så tror vi på redaktionen på denna form av utveckling. Delad glädje är dubbel glädje. När man kan dela med sig av sina kunskaper och byta erfarenheter med andra entusiaster, kan man också få så mycket större nöje av sin VIC-dator.

Varför inte skriva ett brev och dela med Er av Era erfarenheter till en ännu större grupp. Vi på VIC-rapport är intresserade av bidrag.

En möjlighet att dela med sig av sina erfarenheter är ett modem som tillåter kommunikation mellan VIC-datorer. Men hjälp av detta modem kan exempelvis VIC-klubben i Västerås sända meddelanden till VIC-klubben i Haparanda. Detta modem lanseras i dagarna av Handic Software. Kontakta Din VIC-handlare för närmare information.

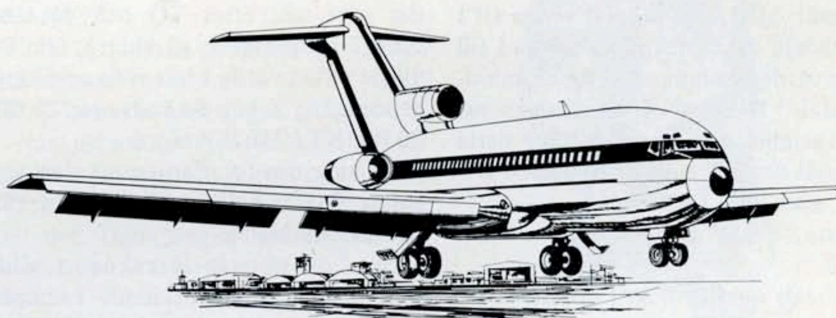
VIC-redaktionen har i dagarna utökats med ytterligare en person; Nina Linander. Meningen är att hon så småningom övertar ansvaret för VIC-rapport. Ni kommer att få höra mer från henne senare.

Innehållsregister

VIC redaktion: Box 42054, 126 12 Stockholm
Ansvarig utgivare: Tomas Wredler
Redaktör: Nina Linander
Övr. medarbetare: Mats Nilsson, Åke Hedman, Joakim Aspengren, Ola Johansson
Annonser: Telefon 08-744 59 20
Tryck: Dala-Offset AB, Falun

Annonsorder och annonsmaterial (heloriginal eller negativ film) enligt överenskommelse under 1984.
Tryckförfarande: Offset

Upplaga 20.000 ex
Nr 3 av VIC-rapport utkommer den 16 april. Manusstopp för nr 4 är den 18 april och annonsstopp den 20 april.



Nybörjarskolan Del 4 VIC-20



I detta nummer skall vi, som utlovat, titta lite på hur man flyttar markören på skärmen samt gör s k loopar och lite annat smått och gott. Som vanligt kan vi inte ta något ansvar för de som följder dessa sidor kan få för den erfarna programmeraren men hoppas att den kan ge något för den verklige novisen.

Vi har tidigare lärt oss hur man skriver (PRINT:ar) olika saker på skärmen, hur man använder ''VICKE'' som räknedosa etc. Vi vet också hur man använder variabler, både strängvariabler och numeriska (tal). I artikel två såg vi också exempel på loopar, låt oss titta lite närmare på dessa.

Loopar

Med en loop menar vi ett programavsnitt som körs ett förutbestämt antal ggr. T ex kan vi tänka oss att vi vill skriva
''HEJSAN''
på skärmen tio ggr.

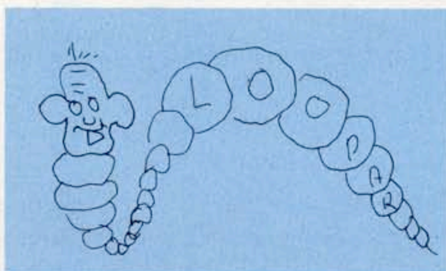
Vi kan skriva detta som

```
10 For I = 1 TO 10
20 PRINT ''HEJSAN''
30 NEXT I
```

I den första raden i programet så talar vi om hur många ggr vår LOOP skall köras, d v s hur många ggr datorn skall utföra det som står på rad 20.

För att tala om detta så måste vi först bestämma oss för att använda en variabel som vi räknar upp varje gång loopen har körts ett varv. I ovanstående exempel har vi alltså valt I för detta ändamål. Med I = 1 så talar vi om att I skall börja räkna från 1. Ändras I till större värde så kommer alltså ett mindre antal ''HEJSAN'' att skrivas ut. Den variabel man använder för detta ändamål brukar man kolla LOOPINDEX, eller bara INDEX.

Nästa ''ORD'' på rad 10 är
TO 10
vilket talar om hur högt Loopindex får ökas innan vi går ur loopen d v s fortsätter med eventuella rader efter rad 30. Ändrar Du tio här till 100 istället så får du alltså 100 ''HEJSAN'' utskrivna på skärmen.



På rad 30 så ligger NEXT I. Detta ord kontrollerar om I är lika stort eller större än det tal vi angav efter TO, efter att först ha ökat I med 1. Är det inte det så sker ett hopp tillbaka till den rad där FOR ligger, och Loopen körd ännu en gång. Någongång så är förhoppningsvis I lika med eller större än det som står efter TO och då låter NEXT programmet gå vidare. Om Du vill se vilket värde I har vid varje loppgenomgång så kan du ändra rad 20 till
20 PRINT ''HEJSAN'', I
d v s tala om för datorn att den förutom ''HEJSAN'' skall skriva ut värdet på variablen I.

Som du noterat så räknas I alltid upp med ett i ovanstående exempel. Detta är inte ett måste, utan man kan ta med ett kommando till för att ange hur mycket I skall ökas med varje gång vi går igenom loopen. Ändra rad 10 till
10 FOR I = 0 TO 20 STEP 3
Om Du nu studerar värdet av I vid varje genomgång av loppen så får Du
0, 3, 6, 9, 12, 15, 18
Innan ''READY'' skrivs ut på skärmen. Med lite enkel huvudräkning kan Du direkt konstatera att I har ökats med 3 för varje varv i loopen. Adderar man tre till arton så får man resultatet 21 vilket är större än tjugo och alltså kommer vi aldrig att få 21 utskrivet. STEP, med ett efterföljande tal, anger alltså hur mycket loopindex skall ökas med efter varje varv i loopen.

Naturligtvis så finns det inget som säger att Du måste använda tal vid definitionerna av hur många ggr din loop skall köras. Det går lika bra att t ex skriva

```
10 X1 = 0
20 X2 = 20
30 X3 = 3
40 FOR I = X1 TO X2 STEP X3
50 PRINT ''HEJSAN'', I
60 NEXT I
```

På detta sätt kan man innan loopen skall köras, bestämma hur många ggr den skall köras, eller ange det värde loopindex skall starta från. Man har också möjlighet att räkna ner i en loop genom att ange ett negativt tal efter STEP
10 FOR K = 10 TO 1 STEP—1
är ett exempel på detta.

I artikel två i denna serie utlovades att vi skulle titta på ett program som skriver ut en multiplikationstabell helt enligt våra egna önskemål. Nu har vi kunskaper att göra en sådan. Ett sätt att göra det på är som följer. Kom ihåg att skriva
NEW
så att det program Du redan har i datorn försvinner innan Du börjar slå in det här.

```
10 PRINT '' DETTA PROGRAM SKRIVER''
20 PRINT ''UT EN VALFRI''
30 PRINT ''MULTIPLIKATIONS-TABELL''
40 PRINT
50 INPUT ''ANGE TABELL''; TB
60 PRINT '' '': REM RENSA SKÄRM
70 PRINT TB; '':ANS''
80 PRINT ''MULTIPLIKATIONS-TABELL.''
```



90 PRINT
100 FOR I = 0 TO 12
110 PRINT TB;" ";I;" = ";TB I
120 NEXT

Programmet behöver väl inte kommenteras vidare.

" " = Rensa skärm erhålls genom att trycka på SHIFT +

Det finns naturligtvis ingenting som hindrar oss från att ha en loop inuti en annan. Följande program skriver ut alla multiplikationstabeller från "ettans" till "tolvans".

20 FOR X = I TO 12
30 PRINT " ";X;"ANS":PRINT"MULTIPLIKATIONSTABELL":PRINT
40 FOR Y = 0 TO 12
50 PRINT X;"*";Y;" ";X*Y
60 NEXT Y
70 PRINT:PRINT"TRYCK PÅ EN TANGENT!"
80 GET AS :IF A = " "THEN 80
90 NEXT X

Det finns ett par saker i detta program som vi måste titta lite närmare på. För det första så måste vi vara noga med att sätta respektive FOR/NEXT sats i rätt ordning. I ovanstående exempel skall den loop som har index Y ligga innanför den som har index X. Detta betyder alltså att NEXT Y måste komma före NEXT X. Du kan ju byta raderna 60 och 90 med varandra för att se vad som händer.

Den andra saken som Du bör lägga märke till är att vi på några rader har skrivit ett ":" och sedan gett ännu ett basic kommando. Detta är ett sätt för att minska ned det antal rader ett program kräver. D v s istället för att skriva

10 PRINT "KALLE"
20 PRINT "KAJSA"

så kan man skriva

10 PRINT "KALLE":PRINT "KAJSA"

vilket ger exakt samma resultat. Vilken metod man föredrar är helt upp till en själv men vill man ha ett lättläst program är nog den första metoden att föredra. Du bör dock veta att man kan spara in lite minne genom att använda sig av den andra metoden men att detta i de flesta fall varken gör till eller från.

Den tredje kommentaren kretsar kring rad 80. I denna rad hittar Du tre nya basic kommandon.

GET
IF
THEN

Den första av dessa (GET) tar in ett värde från tangetbordet i den variabel som följer den. Du kan skriva in följande program för att se hur det här fungerar.

10 GET AS
20 PRINT AS
30 GOTO 10

GOTO i detta program gör precis vad den ser ut att göra d v s "GÅR TILL" den rad som anges.

När programmet ovan körs så ser det först ut som om ingenting händer. Tryck på en tangent så får du se. Ja just det. Den tangent Du trycker på skrivs ut på skärmen.

Med
GET AS

så får alltså AS värdet, bokstaven/siffran, som man trycker ned. OBS! dock att AS även får ett värde när du *inte* trycker ned en tangent. Detta värde kallar man TOMSTRÄNG och är precis samma som ges när PRINT " " körs. Att detta är ett VÄRDE för variabeln AS kan kanske tyckas vara lite underligt men man kan tänka sig att en TOMSTRÄNG representerar samma sak för strängar, som en nolla gör för siffror.

Instruktionerna IF/THEN skall vi titta lite mer på senare men vi kan avslöja redan nu att de används för att testa om något är SANT och utför detta NÅGOT bara Om det vi testat är sant. Det som skall testas skall stå efter IF och det som skall utföras, om SANT, skall stå efter THEN.

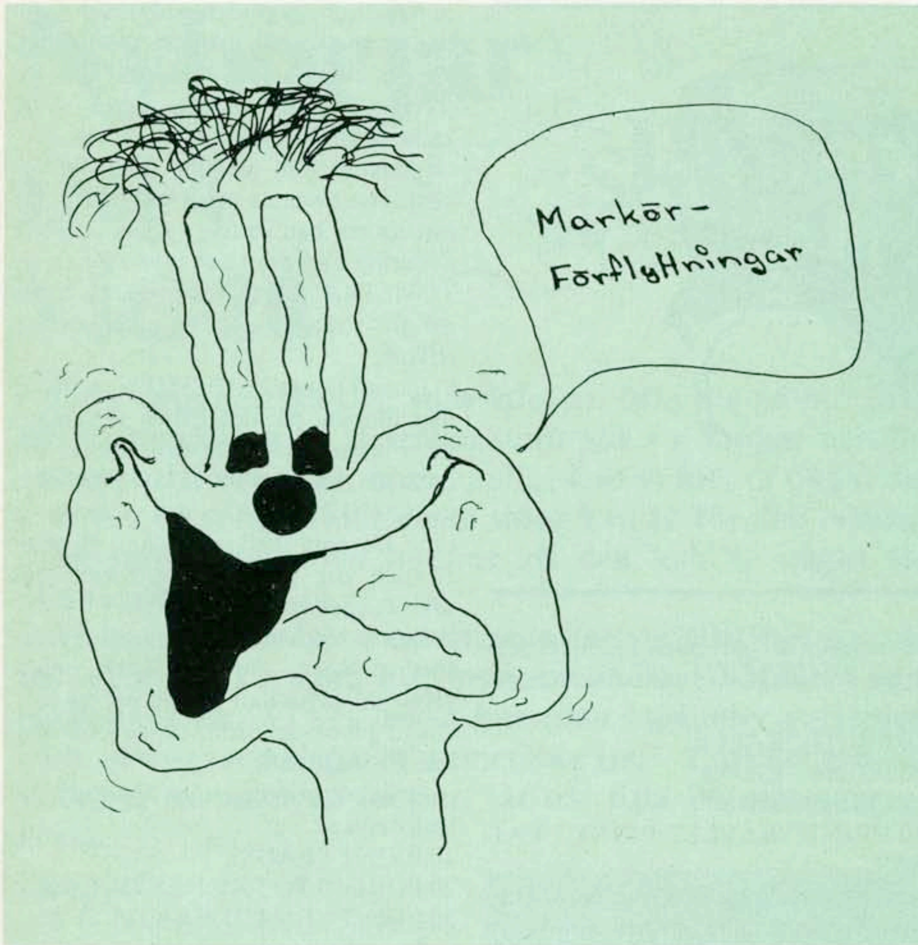
Och så var det så den utlovade tidsfördröjningen. Du har säkert redan gissat att detta kan åstadkommas helt enkelt genom att ordna till en loop som inte gör något alls.

Följande programsnutt är ett sätt att lösa detta på.

10 PRINT "PAUS"
20 FORI = 0 TO 5000:NEXT
30 PRINT "EN TID HAR GÅTT."

Som Du kanske noterat så har jag ibland utelämnat I, eller vad nu loopindex är, från NEXT kommandona. Detta går alltid bra att göra. Även i de fall man har flera loopar i varandra. En nackdel har det dock. Man ser inte lika lätt i sitt program var den ena loopen börjar och den andra slutar. Så till en början, innan Du är riktigt van kan det löna sig att sätta ut dessa.





Markörförflyttningar

Genom att trycka på CRSR tangenterna på tangentbordet kan Du som Du sett, och säkert använt dig av åtskilliga ggr, flytta omkring markören på skärmen. Vad Du kanske inte vet är att du även kan använda dessa i dina PRINT kommandon. Detta Är mycket användbart i många fall och kan t ex vara nyttigt att kunna när man skall göra spel och liknande. Det är dock inte bara dessa tangenter Du kan använda utan även andra tangenter som påverkar skärmen på något vis går bra. Låt oss göra en lista över dem:

Shift + CLR/Home

Flyttar upp markören till översta vänstra hörnet efter att först ha raderat hela skärmen.

Inom situationstecken skrivs

CLR/home

Flyttar upp markören till översta vänstra hörnet av skärmen utan att radera den.

Inom situationstecken skrivs

INST/Del

Tar bort det tecken som står före markören.

Inom situationstecken skrivs

CRSR Upp

Flyttar markören uppåt en rad. Inom situationsteckens skrivs

CRS Ner

Flyttar ned markören en rad. Inom situationstecken skrivs

CRSR Vänster

Flyttar markören ett steg åt vänster. Inom situationstecken så skrivs

CRSR Höger

Flyttar markören ett steg åt höger. Inom situationstecken så skrivs

Med ÅÄÖ installerat

De tecken som skrivs ut när man trycker på en av dessa tangenter kallas KONTROLL TECKEN.

Användning av SHIFT + CLR/HOME har vi redan sett användas i de exempel vi tittat på i detta nr. Förutom att använda dessa tangenter så kan man också använda tangenterna för att byta färg och få reverserade tecken alltså dem man kan hitta under siffrorna. Testa dessa i olika printkommandon. Ha roligt! Du kan ju inte förstöra något, så vad gör det om något blir fel. Ha det så bra tills nästa månad då vi skall titta på nya SPÄNNANDE saker Du kan använda "VICKE" till.

```
5   PRINT"(SHIFT + CLR/HOME)(CRSR NER)(RVS ON)(22 blanktecken)(RVS OFF)"
10 PRINT"(SHIFT + CLR/HOME)"
20 FOR I = 1 TO 20
30 PRINT"(SHIFT + CLR/HOME)"
40 FOR J = 0 TO I
50 PRINT"(CRSR HÖGER)";
60 NEXT J
70 PRINT CHR (65 + I); :FOR K = 1 TO 90: NEXT:PRINT
80 NEXT I
90 PRINT"(SHIFT + INST/DEL)"
100 GOTO 5
```

Det som står inom parenteser skriver Du inte in i detta program utan *gör* det som står inom prentesen istället. T ex i fallet (SHIFT + CLR/HOME) så håller Du alltså SHIFT-tangenten nedtryckt medan Du trycker på CLR/-HOME-tangenten. Programmet presenterar inte särskilt mycket men kan, förhoppningsvis, introducera dig i markörförflyttningen värld.



HEJ!

Biforgar programlista till ett bollspel för Commodore 64. Det kan med små ändringar köras även på VIC 20.

Det gäller att plocka så många poäng som möjligt. Spelet har 4 nivåer. På de första (upp till 300 resp. 3000 poäng) har man en bredare 'spade' att studsa bollen med, — över 3000 blir 'spaden' smalare, då den horisontella delen försvinner. Fjärde nivån uppnås på 6000 poäng.

Bollen studsar mot väggar och tak, och när den träffar 'spaden' studsar den 'logiskt' åt det håll som ytan lutar. När ett tecken plockas i raderna upptill på skärmen, blir riktningen på bollen slumpvis rakt ner eller snett ned åt höger eller vänster. Ett ljud avges vid varje studs eller teckenplockning.

Bommar man bollen med 'spaden' förloras bollen. Tre nya bollar fås för varje nivå som uppnås, och ev. kvarvarande bollar från tidigare nivåer får behållas. När alla bollar missats är spelet slut. En ringklocka ringer och spelet startar på nytt när mellanslagstangenten trycks ned.

Maximalt uppnådda poäng sparas givetvis och visas hela tiden vid sidan av aktuell poäng.

För den som vill skriva om spelet till VIC 20 gäller att VIC 20 har mycket enklare ljudkommandon, varför rad 22 (som nollställer hela ljudregistret) kan hoppas över, och rad 370—410 ska ändras.

På VIC 20 kan tecknen skrivas ut över hela skärmbredden (ändra rad 500). RF på rad 10 är början av färgminnet.

När du tryckt in programmet, så spara det först på band innan du provkör. Mängden POKE-kommandon bäddar annars för att datorn hänger sig om du gjort något fel.

Mycket nöje!

Birger Gran

```
6 PRINT"▓":PRINTTAB(9)"▓▓▓▓▓* * BOLLSPEL * * *":REM AV BIRGER GRAN
7 PRINTTAB(12)"▓▓STYR PLATTAN MED":PRINTTAB(12)"▓CRSR-TANGENTERNA"
9 P(1)=300:P(2)=3000:P(3)=6000:TE(1)=65:TE(2)=83:TE(3)=88:R=1024:LJ=54272
10 A=0:TE=90:PD=10:BO=0:RF=R+LJ:SV=0:H=0
20 PRINTTAB(10)"▓▓▓▓▓▓▓START TRYCK <SPACE>":IFP>PMTHENPM=P
22 FORI=LJTOLJ+24:POKEI,0:NEXT:POKELJ+24,15:POKELJ+5,8:POKELJ+6,8
23 FORLL=1TO25:FR=6:GOSUB370:NEXT
24 GET A$:IFA$<>" "THEN24
25 PRINT"▓";:PRINTTAB(28)"MAX:";PM:GOSUB500
26 FORT=2TO22:PRINTTAB(8)"X"SPC(21)"X":NEXT:P=0
32 H=H+1:IFH>2THENSV=1
34 PRINT"▓"SPC(17)"        "
35 FORQ=1TO(3+BO):POKER+17+Q,81:POKERF+17+Q,7:NEXTQ
36 FORW=1TO(3+BO):POKER+17+W,32:S=1913:SF=56176
40 FORT=8TO32:POKESF-1+T,10:NEXTT:V=S:GOSUB300
50 X=20:Y=12:DX=-1:DY=1:XY=R+X+40*Y
80 POKEXY+LJ,10:POKEXY,81:POKEJ,32:J=XY
110 X=X+DX:IFFR>0THENGOSUB370
112 IFX<9THENX=9:FR=1:DX=-DX
114 IFX>29THENX=29:FR=1:DX=-DX
134 Y=Y+DY:IFY>24THENFR=5:GOSUB310:POKEJ,32:NEXTW:GOTO10
140 IFY<1THENY=1:FR=3:DY=-DY
200 XY=R+X+40*Y:PE=PEEK(XY):IFPE=32THEN274
220 IFPE=TETHENDY=-DY:P=P+PD:DX=INT(RND(1)*3)-1:FR=2
225 PRINT"▓   POANG:";P
230 IFP=P(H)THENGOSUB310:TE=TE(H):PD=50*H:GOSUB500:P=P+PD:BO=4+BO-W:GOTO32
250 IFPE=78THENDX=-1:DY=-DY:FR=4:GOTO110
260 IFPE=99THENDX=0:DY=-DY:FR=4:GOTO110
270 IFPE=77THENDX=1:DY=-DY:FR=4:GOTO110
274 IFPEEK(197)=2THENV=S+1:IFV<1934THENGOSUB310:GOSUB300:S=V
276 IFPEEK(197)=7THENV=S-1:IFV>1910THENGOSUB310:GOSUB300:S=V
290 GOTO80
300 POKEV,78:POKEV+1,99:POKEV+2-SV,77:RETURN
310 FORI=0TO2:POKES+I,32:NEXT:RETURN
370 ONFRGOTO372,374,376,378,380,390
372 FH=32:FL=94:GOTO405
374 FH=64:FL=188:GOTO405
376 FH=45:FL=198:GOTO405
378 FH=8:FL=23:GOTO405
380 FH=108:FL=223:GOTO405
390 FH=183:FL=25
405 POKELJ+1,FH:POKELJ,FL:POKELJ+4,17
410 FORT=1TO25:NEXT:POKELJ+4,16:FR=0:RETURN
500 FORD=2TO4:FORE=9TO29:POKER+E+40*D,TE:POKERF+E+40*D,10:NEXTE,D:RETURN

READY.
```

Grattis på 25-årsdagen



COMMODORE!

Ett kvarts sekel med Commodore: Från detaljist till världsomfattande jätte.

På bara 25 år har en liten skrivmaskins och reparationsrörelse undanskymt lokaliserad i kärnan av staden Toronto, Canada, utvecklats till en av de mest intressanta personadatorföretagen i världen. Commodore International Limited.

Företaget levererar mer produkter runt om i världen än något annat dataföretag. Commodore har växt från en omsättning på 46 miljoner dollar 1977, till över 680 miljoner dollar 1983, (bokföringsåret slutade 30 juni). Mycket av framgångarna har att göra med den entreprenörs anda som präglar Commodores grundare och nuvarande vice ordförande, Jack Tramiel.

Den polskfödde Tramiel överlevde nazisternas koncentrationsläger under andra världskriget. 1958 emigrerade han till Nordamerika och öppnade sin egen skrivmaskins affär i Toronto. Tramiel har alltid haft ''näsa'' för vad som efterfrågas i framtida hem och företag i elektronikväg. Dessutom har han snabbt haft möjlighet att tillgodose sina kunders önskemål. Commodores framgång bygger på denna affärsfilosofi.

Under 25 år har Tramiel ledsagat Commodore från mekaniska räknemaskiner, miniräknare, digitala armbandsur fram till persondatoråldern. Tillsammans med kunnig företagsledning runt om i världen försöker Tramiel utröna: vad väntar härnst? Commodore är erkänt för sitt enkla men geniala sätt att agera ''håll fast vid det gamla så länge det är bra, byt till det nya så fort det blir bättre än det gamla.

Under tidiga år växte Commodore från skrivmaskinsreparatör och försäljare till skrivmaskinstillverkare genom förvärv av en fabrik i Berlin, Västtyskland. I början av 60-talet började Tramiel sälja och bjuda ut en stor mängd olika kontorsutrustning och dessutom nationellt distribuera kontorsmöbler åt ett företag i branschen.

1965 köpte Commodore möbeltillverkaren och flyttade sin verksamhet till vad som nu är Commodores nuvarande kanadensiska huvudkontor. Företaget tillverkar fortfarande kontorsmöbler (som huvudsakligen passar till företagets persondatorsortiment). Vid sin fabrik i Scarborough, Ontario. Hela kontorsmöbelverksamheten har expanderat till tre kontor och två tillverkningsenheter i Torontos grannskap.

Under år 1965 träffade även Tramiel den kanadensiske advokaten och finansiären, Irving Gold, som senare blev Commodores styrelseordförande. Dessa två tillsammans byggde upp och formade företaget till vad det är idag. En av de första saker de gjorde var att sälja Commoderes mekaniska räknemaskinsfabrik och istället försöka hitta ett Japanskt företag som kunde tillverka räknemaskiner som Commodore istället kunde distribuera. När Tramiel var i Japan och för första gången fick syn på elektroniska kalkylatorer, insåg han snabbt att den mekaniska räknemaskinens framtid var förbi. Av detta drog Tramiel slutsatsen: tar vi inte upp den här produkten nu kommer någon annan göra det före oss. Tramiel agerade snabbt och fann snart en tillverkare av elektroniska räknare som var villig att tillverka produkterna med Commoderes namn. På grund av detta var företaget redan med från början i den nya branschen.

Företaget började tillverka sina egna elektroniska kalkylatorer 1969, med Texas instruments komponenter (chips). Faktum är att Commodore var det första företaget på marknaden som började sälja s k ''mimiräknare'' i fickformat —c108:an—, som är ett exempel på vad som blivit Commodores riktmärke, vara först på marknaden med något nytt.

Intressant att notera är att denna produkt såldes till i stort sett samma pris, genom samma distributions kostnader och till samma kunder som till den populära VIC-20 datorn idag gör.

Fram till 1974 utökade Commodore sitt sortiment från enkla fyra funktionsräknare, vetenskapsmodeller och programmerbara tangentbordsmodeller. Vid denna tidpunkt var företaget mycket beroende av en tredje part — de som tillverkade ''chips'' och lysdiodrar som ingick i Commodores produkter.

1975 beslutade sig Texas Instrument för att gå emot sina egna kunder, genom att själv börja tillverka elektroniska kalkylatorer. Samtidigt med detta sjönk ''chip'' priset från 12 till en dol-

lar. Detta medförde att Commodore satt fast med ett stort lager av ''chips'' och räknare när marknadspriset på produkterna kraftigt sjönk. Det var vid denna tidpunkt Tramiel bestämde sig för att företagets utveckling inte skulle ligga i andra företags händer, utan Commodore skulle i större utsträckning styra sitt eget öde.

Kort därefter, 1976 köpte Commodore MOS Technology Inc en av företagets leverantörer av ''halvledare''. Detta för att i större utsträckning bli vertikalt integrerade. Detta har i sin tur fört med sig att Commodore nu är självförsörjande på komponentsidan, som i sin tur medför att företaget har betydligt längre s k ledtid (från produktutveckling till kommersiell lansering) för sin utveckling av nya produkter. Detta faktum leder till kostnadsfördelar, som i sin tur visar sig i produkter till lägre pris och bättre prestanda.

Köpet av MOS Technology följdes av ytterligare två förvärv inom loppet av 18 månader: köpet av Frontier, en Los Angeles tillverkare av ''chips'' som kompletterade de från MOS, och förvärvandet av Dallas baserade Micro Display System Inc. En tillverkare av ''liquid crystal display'' utrustning. Resultatet av dessa företagsköp innebar att Commodore hade bättre intern expertis och produktion i flera nyckelteknologier än något annat större företag i branschen.

Under 1976 omorganiserade Commodore sin samfällda struktur — Commodore International Limited — och flyttade sitt finansiella huvudkontor till Bahamas, och sitt verkställande högkvarter till Wayne i Pennsylvania (det har sedan dess flyttat till West Chester i Pennsylvania).

Näst förjande år var ''avstampet'' för Commodore — som fortfarande deltog i framtidsutveckling i gammal god Commodore stil — då företaget introducerade sin första persondator, PET:en. PET:en (Personal Electroic Transactor) använder MOS-tillverkade 6502 microprocessorer som också används av en del av Commodores konkurrenter. Det var denna orginal maskin, som introducerades på Hannovermässan i Tyskland och vid konsument elektronik-mässan i USA, som ''döpte'' persondator marknaden av idag.

Sortering med Basic — UTAN VÄNTETID!

I anslutning till artiklarna om sortering i VIC-rapport nr 5/6, vill jag bekräfta att väntetiden vid sortering av större datalistor är oacceptabelt lång med program skrivna i BASIC.

Jag stötte på detta problem när jag arbetat med ett statistikprogram till Commodore 64-an. Där behöver datalistan sorteras för att få fram medianvärdet samt för frekvenstabellerna. Väntetider på en halv timme för 2—300 poster är avskräckande — och att sortera 1000 poster (som programmet i övrigt klarar av att bearbeta) vågar jag inte ens tänka på, eftersom tidsåtgången ökar exponentiellt.

Lösningen blir i stället följande princip: utnyttja datorns överlägsna hastighet under inmatningsfasen. Tekniskt går det till så att en parallell matris läggs upp dit de inmatade posterna direkt placeras i storleksordning.

För de beräkningar som fordrar en sorterad lista används B-matrisen och för korrelationsanalyser m m används den ordinarie A-matrisen.

Om datalistan ska sparas i datafil på band, måste man naturligtvis tänka på att spara både A- och B-matrisen, då vitsen med storteringsrutinen annars går förlorad. Detta tar dock ej påtagligt längre tid.

Här följer en utskrift av ett programexempel, som sorterar valfri storlek av datalista (upp till 2000 poster) och sedan skriver ut originallistan följd av en utskrift av den sorterde listan.

Birger Gran

PS. Återkommer senare med en beskrivning av statistikprogrammet, som kanske är för långt att publicera, då det tar 7 sidor på printern. Annars kanske det har ett allmänt intresse, eftersom nyttoprogrammen fortfarande är alltför få till hemdatorerna. DS.

```
SORTERINGSRUTIN

5 REM SORTERINGSRUTIN AV B. GRAN
7 PRINT"♥":N=0:DIMA(2000),B(2000)
8 N=N+1:T$="":INPUTT$:IFT$=""THEN15
9 T=VAL(T$):A(N)=T
10 X=0:FORI=1TON-1:IFT<B(I)THENX=I:I=N-1
11 NEXT
12 IFX=0THENB(N)=T:GOTO8
13 FORJ=NTOX+1STEP-1:B(J)=B(J-1):NEXT:B(X)=T:GOTO8
15 N=N-1:PRINT"♥INMATAD LISTA":FORI=1TON:PRINTA(I);:NEXT
16 PRINT:PRINT:PRINT"SORTERAD LISTA"
17 FORI=1TON:PRINTB(I);:NEXT

READY.
```

Äntligen har Electronic Arts programutbud kommit till Sverige genom Handics försorg. Extremt välgjorda program, snyggt förpackade och utförligt dokumenterade. Jag har haft turen att få testa två stycken, nämligen M.U.L.E. och Music Construction Set.

M.U.L.E.
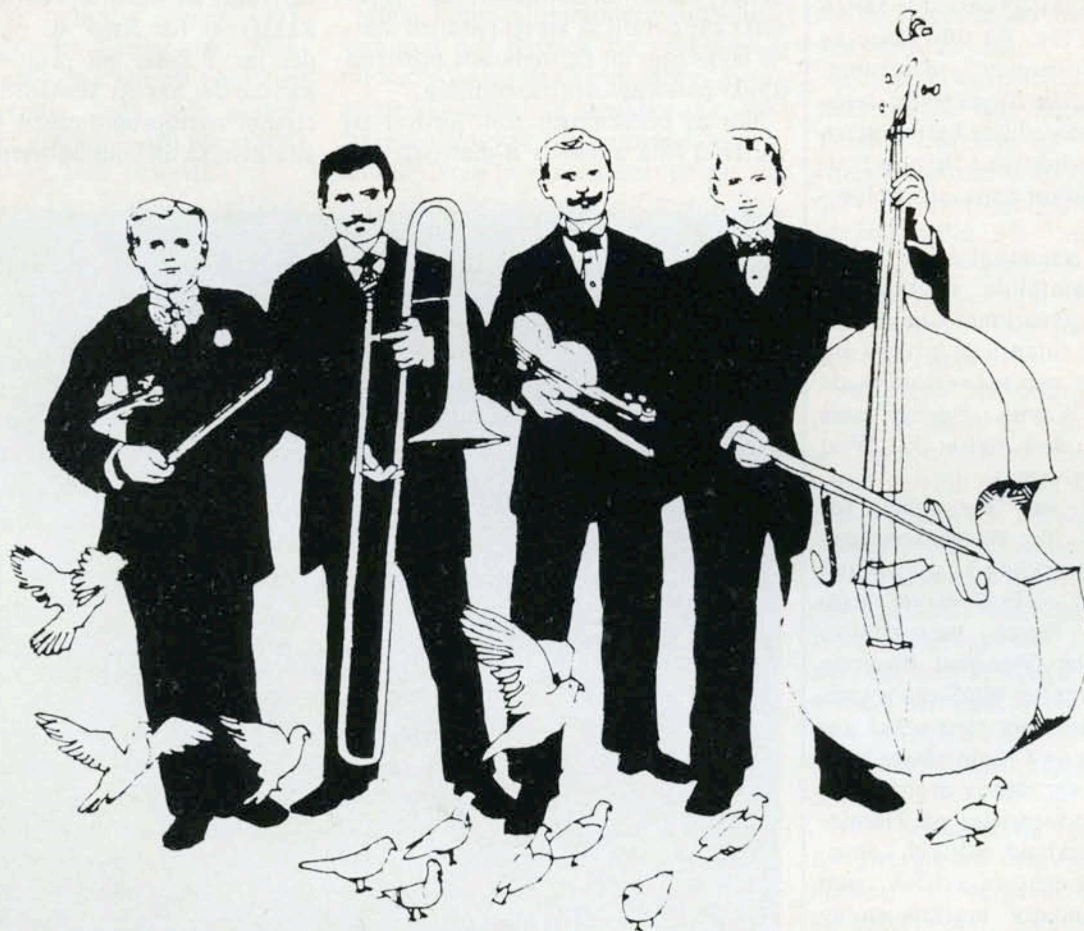
står för ''Multiple Use Labor Element'', fleranvändbart arbetskraftselement alltså, men ordet 'mule' betyder mula, mulåsna. Det är med hjälp av dessa bråkiga, bångstyriga mulor som du ska kolonisera planeten Irata och försöka tjäna så mycket pengar som möjligt på 6 eller 12 månader.

En typisk spelomgång kan se ut så här:

Speldisketten stoppas in och du laddar in programmet ''EA'' som i sin tur laddar in själva spelprogrammet medan Electronic Arts tjusiga logotyp visas på skärmen, konstant ändrandes färg. Under tiden kan du bläddra lite i regelhäftet och studera sidorna med specialtips om spelet eller läsa vad de fyra spelkonstruktörerna har för taktik. Programmet startar automatiskt med att visa en liten tecknad film och spela tjusig musik, originalskriven av Roy Glover. Trummor, trestämmigt och lika pampig som Star Wars melodin! Du kan välja mellan 3 svårighetsgrader Beginner, S-tandard och Tournament samt antalet personer från 0 (!) till 4. Det är alltid 4 spelare med i spelet, de som inte styrs av mänsklig hand tar datorn kontroll över. Om du väljer 0 spelare kör datorn alltså en demorunda där han själv tar hand om alla spelarna. Skillnaden mellan de olika svårighetsgraderna är bl a hur många omgångar man ska spela, hur snabbt utdelningen av land går, möjligheten att gräva efter Chrystite, att ha auktion på land m m. Mer om detta senare.

Nu till en verkligt smart sak: val av joystick/tangentbord. Datorn uppmanar alla deltagare att trycka på sin fireknapp och avgör vilken joysticksport som 'pinnen'' sitter i eller (om man är flera spelare) om man vill använda tangentbordet. Så borde det vara på flera spel tycker man för vartannat spel använder ju port 1 och vartannat port 2, ofta utan att ens tala om vilket. Först när rymdskeppen kommer susande mot dig märker du att du har joysticken i fel port! Men tillbaks till spelet.

Varje spelare får välja färg och spelarfigur. Åtta olika varelser med olika namn, egenskaper och hemplaneter finns till förfogande, däribland en människoliknande för experter och en ''f-lapper'' för nybörjarna. Medan man väljer får man se dem framifrån bakifrån och från sidorna. Till sist är



vi redo att landa på planeten och börja spela.

Rymdskeppet landar i den lilla staden och ut tumlar vi 4 glada kolonisatörer, redo att börja odla marken i floddalen, sätta upp våra solfångare på slättlandet och börja med lite gruvdrift i bergen. Men vad allt är dyrt! Nåja, vi får ju ett litet startkapital och marken är gratis. En svart fyrkant glider över landskapet och när den är över en plätt som verkar bra trycker man på fireknappen. Har man tur och ingen annan hinner före så får man en tomt att bruka, men först måste du bebygga tomten. När det blivit din tur att gå på en shoppingrunda i staden så går du och köper dig en mula i inhängnaden i stadens östra del. Finns det gott om mulor är priset lågt och vice versa. Välj sedan hur du vill utrusta mulan och gå in i rätt affär, spring till din landbit och tryck på fireknappen. Vips, så är mulan förvandlad till en jordbruks-, energi-, eller gruvenhet!

Har du gott om tid (vilket beror på hur mycket mat du har i lagret) kan du försöka fånga den mystiske Wampusen som bor i hålor i bergen. Om du får tag i honom så betalar han en ordentlig lösensumma. Om du vill kan du istället springa tillbaks till staden och gå på Puben där du vinner en slant på spel, ju mer tid kvar desto mer vinner du men mest är det slumpen som avgör. När din tid tagit slut (eller om du rusade in på Puben) så är det näste mans tur.

Efter det att alla är färdiga så startar produktionen som styrs av tomternas lämplighet, varje spelares energiförråd och naturligtvis slumpen. Olika naturkatastrofer sätter hinder i vägen, bl a pest, planetbävningar, solfläckar, meteoriter o s v. Nu kommer den roligaste biten: auktionerna.

De som har överproduktion säljer till högsta möjliga pris och de som inte har tillräckligt försöker köpa billigt, allt helt visuellt genom att säljarna är överst i bild och 'går ner' i pris medan köparna startar i underkant och går uppåt. Detta köpslående kan bli verkligt roligt om man är några stycken som kan skrika ut hotelser, löften och förbannelser medan buden höjs och sänks.

Efter det att ställningen har visats börjar nästa omgång med nya utmaningar, beslut, bekymmer och framgångar. Den som efter 6 eller 12 spelomgångar har mest tillgångar i form av pengar, land och varor vinner, men hela kolonin måste ha en viss gemensam vinst för att ha 'lyckats'. Samarbete är alltså nödvändigt. Nå, hur bra är det här spelet då? Svar: mycket, mycket bra. Roligt, omväxlande, krävande, spännande, välgjort, realistiskt. Musik, grafik, regler, allt är i toppklass. Särskilt roligt är det om man är 2 eller flera spelare men man kan även spela själv mot datorn. Många drag har lånats från vanliga sällskapsspel, typ Monopol, men anpassasts till datorn. De många olika slumpmomenten som alltid är bra för den som ligger sist och dåliga för den som leder tillsammans med handikappsystemet med olika figurer gör spelet jämnt. Svårighetsgraderna är lämpligt avpassade för att föra in nya moment som gör spelet mer komplicerat när man lärt sig de enklare transaktionerna. Finns det då inget negativt att säga om M.U.L.E.?



Jodå, några svaga punkter märker man, exempelvis så måste man stänga av datorn och ladda in programmet på nytt om man vill byta färg/figur/antal spelare. Inte så bra. Man får heller inte se hur datorn går runt i staden och köper mulor och utrustar dem. Det bara plingar till så har han ett solkraftverk här och en gruva där. Han fuskar inte sägs det, men vem vet?

Sammantaget, ett jättebra spel som inte bör saknas hos någon spelgalning. Bör kunna locka hela familjen/gänget till datorn, även de som tycker Jumpman och Frogger är för svåra att spela. Och priset är ju rätt lågt, jämförelsevis, bara 375:—. Kräver VIC 64, 1541 disc drive, monitor, joystick.

Music Construction Set

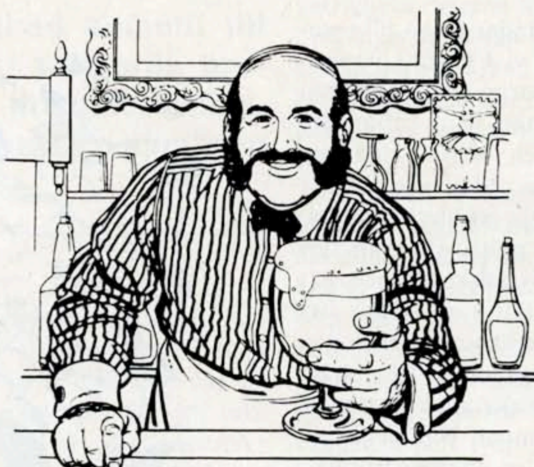
är skrivet av Will Harvey, 16 år (!) och handlar alltså om musik som namnet antyder. Med hjälp av detta program ska du kunna författa musik, koppla ihop en bunt noter med en vacker melodi samt testa olika rytmer och klanger.

Programmet EA laddas in och det laddar i sin tur in huvudprogrammet på samma sätt som M.U.L.E. Ett notsystem med melodi- och basstämma skrivs ut med ungefär 2 takter synliga. Under notsystemet till vänster finns allehanda noter, pauser och förtecken. I mitten finns indikatorer för tonart, takt, hastighet, ljudform och volym samt en takträknare. Till höger finns olika symboler, bl a ett piano (spela), ett hus (home d v s utgångspositionen), sax och lim (editering) och en diskett (för diskoperationer). Om du inte trycker på RETURN börjar en demorunda som spelar den 13 låtar som följer med på disketten, samtidigt som noterna susar förbi på skärmen.

Annars dyker en liten hand upp som

du styr med din joystick, plockar upp noter och placerar dem i notsystemet, ändrar takt och ljudform, provspelar och till sist sparar stycket på diskett.

Detta verkar ju enkelt och behändigt i teorin men hur har konstruktören lyckats i praktiken? Nåja, det är ju tjusigt med detta ''LISA-liknande'' arbetssätt men för att det ska fungera krävs nog en 'mouse' eller åtminstone en proportionerlig joystick. Det är väldigt segt om man ska flytta 16 sextondelar från nedre vänstra hörnet till notsystemet uppe till höger. Tangentbordsfunktioner finns inbyggda som gör inknappning av melodier möjlig men då har ju hela finessen med det visuella notssystemet försvunnit.



Dessutom blir det lätt grötigt på skärmen vilket gör det mycket svårt att peka på rätt not som ska flyttas och att sedan placera den på rätt ställe är inte heller lätt.

Nu låter det som om Music Construction Set är dåligt men så är inte fallet. I samma klass som M.U.L.E. är det inte riktigt men det är ett väldigt proffsigt program som man kan ha mycket roligt med. För musikkunniga är det säkert användbart eftersom man kan transponera till önskad tonart, ändra takt, pröva olika komp och få noterna utskrivna på skrivare. Det sistnämnda går dock så långsamt att man nästan aldrig gör det och dessutom måste trycka på en tangent efter 2 1/2 takt för att utskriften ska fortsätta. Orsaken till dessa båda olägenheter är antagligen att det handlar om ren kopiering av högupplösningsskärmen, en s k hardcopy, istället för en speciell rutin för skrivarutskrift.

Två andra finesser som åtminstone jag gärna såg är att kunna sätta attack/decay/release/sustain, helst med grafisk uppritning. Jag tycker det är jobbigt att tänka hur ett ljud är uppbyggt om man inte har ett litet diagram att titta på. En möjlighet att spela melodier även i egna program skulle vara användbart. Jämför Pinball Construction Set här nedan som har en Make Game möjlighet.

Slutligen vill jag säga att ljudet är mycket bra, särskilt om man kopplar datorn till en stereoanläggning, och att dokumentationen håller samma höga klass som Electronic Arts övriga produkter. Relativt lågt pris på 375:— gör Music Construction Set till ett mycket intressant program trots att det inte är helt utan brister och nackdelar.

Kräver VIC 64, 1541 disc drive, joystick.

Stefan Andersson

Pinball Construction Set

Ett helt outstånding program från Electronic Arts. Mycket mer än så här kan man nog inte få ut av 64:an. Man ska alltså konstruera och spela flipperspel och när man är färdig kan man göra ett spel och ge bort till en kompis eller sälja. När du laddar in programmet visas en höggrafikbild med Electronic Arts logo med färgändringar. Sedan får du upp på vänster halva en tom flipperspelsram samt på höger halva en mängd delar, t ex bumpers, flipperarmar, bollfångare, magneter.

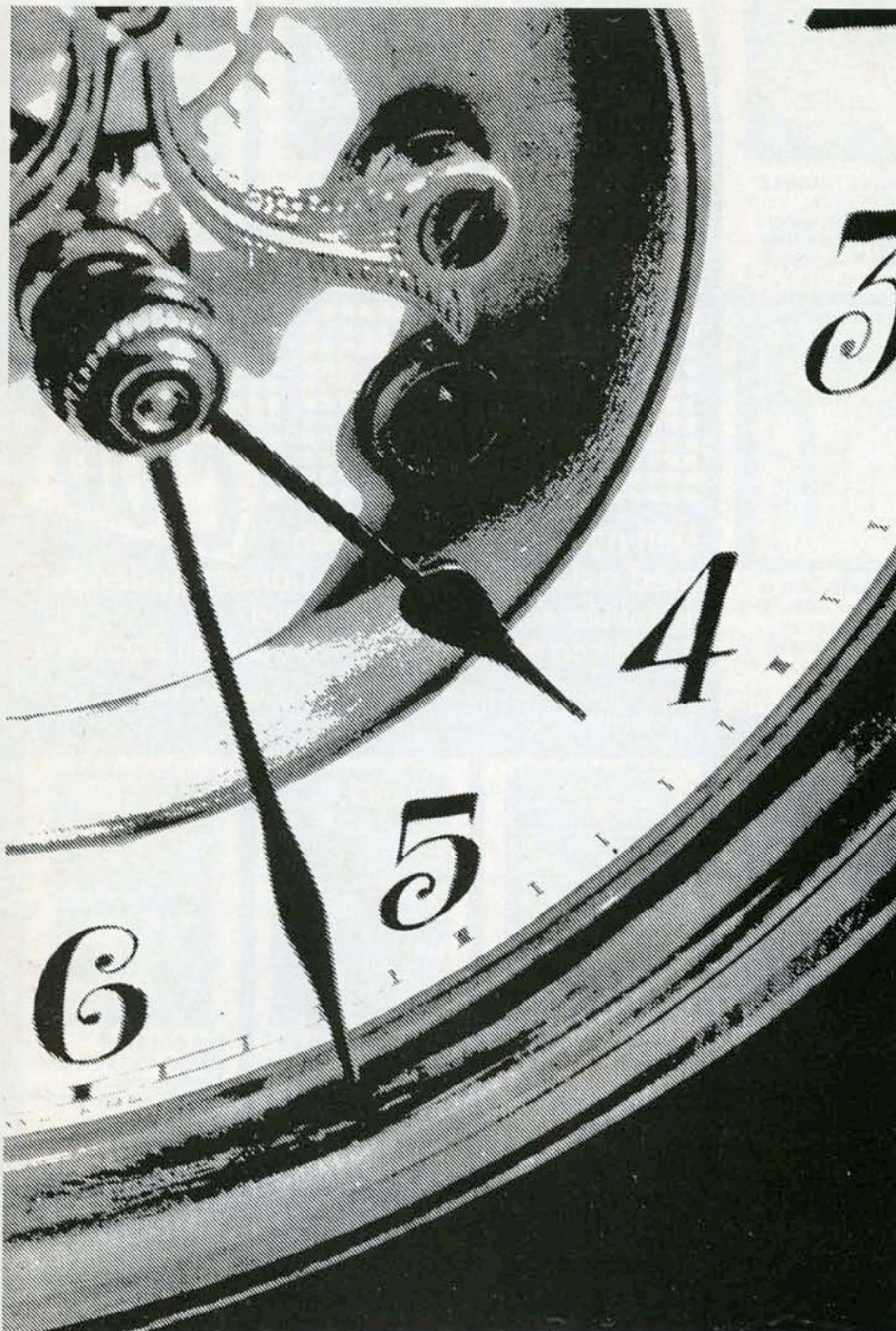
Du för en liten hand och tar tag i något som du vill ha med i spelet och drar ut det till spelbrädet. Du kan även dra fram en fyrkant som du sedan kan forma precis som du vill. Många finesser, du kan förstora en del av skärmen och ändra varje liten punkt, du kan provspela, du kan koppla ihop flera saker så att det ger bonus när du träffat allihop samt ändra fart, gravitation och studs. En fin sak är möjligheten att spara ett komplett spel som kan spelas utan PCS som du kan ge bort till folk.

Omslaget är bra, först tror man att det är en maxisingel men skenet bedrar. Det medföljer en instruktionsbok som är toppen + en minneslapp som också är bra att han och så står det om honom som konstruerade spelet och hans bakgrund.

Det enda negativa är ljudet, synthezisern utnyttjas inte fullt ut, men annars är det ett bra spel som man inte tröttnar på i första taget. Ett av de mest prisvärda programmen på marknaden.

Kräver VIC 64, 1541 disc drive, joystick.

Joakim Aspengren



Miljoner och . . .

(av Jim Gracely)

Tack vare modern teknologi arbetar Din dator med mätningar som rör sig i storleken "en miljon". Det är en punktlig klocka som tickar över åtta miljoner gånger per sekund och med "chips" med komponenter, inte större än 50 miljondel av en meter. I den här artikeln undersöker vi måtten tid och storlek i förhållande till en 6502-"chip" (kiselbricka).

Stora tal är både mystiska och fascinerande. Liknande förhållande råder för små tal. När vi vill beskriva någonting större än någonting annat använder vi uttryck som "miljoner": "Det är en chans på en miljon att Du gissar rätt", eller "Det måste finnas miljontals av dem". Så fort talet blir riktigt stora förlorar vi fattningen om hur stora de egentligen är.

Ett intressant exempel är att testa Din förmåga att förstå stora tal. Utan att använda kalkylator eller dator, försök att välja vilket av följande exempel som är mest pengar.

1. Två ton guld (marknadspriset 11 oktober 1983).
2. En "tio-centare" per inch (2,54 cm) mellan New York och Los Angeles.
3. En "fem-centare" per yard (0.9 m) mellan jorden och månen.

I detta exempel är måttenheterna mycket större än vad Du brukar komma i kontakt med. Den största summan pengar i exemplet ovan är 25.800.000 dollar. Jag kommer längre fram i texten tala om vilket av exemplen som motsvarar beloppet. Hur mycket är 25 miljoner dollar? Om du skulle använda pengarna att köpa bensin för, skulle Du kunna fylla en 15-gallon-tank (ca 57 l) varje dag i 3.802 år.

Tid

Hur relaterar man tid till en mikrodator? Kristalluret i VIC 20 och VIC 64 har en frekvens på 7 MHz. Det betyder att den "tickar" en miljon gånger per sekund. Ett flertal maskinspråk (som t ex INX och CLC) gör beräkningar inom loppet av två "tick", d v s det tar två miljonderal av en sekund att genomföra en beräkning.

Nu ska vi förändra måttenheterna i datorns klocka och göra varje "tick" till en sekund (multiplicera klockan med en miljon). På så sätt kan man bättre illustrera hur lång tid en dator tar på sig för att göra beräkningar. Det som tar datorn en sekund att räkna ut, tar 278 timmar eller 11,5 dagar att räkna ut för hand. Här följer en tabell som jämför verklig tidsåtgång jämfört med hur många "tick" och tidsåtgång med den nyss beräknade modifierade klockan.

Tabellen ovan ger en lite bättre insikt hur snabb en microprocessor är. Här kommer en annan fråga till Dig: Om Du skulle arbeta 11,5 dagar i sträck med bara en halvtimmes rast var

Verklig händelse	Hur många "tick" som går åt på en 1 MHz klocka	Tidsåtgång om tiden är 1 tick = 1 sekund
1 miljondels sekund	1	1 sekund
1 sekund	1.000.000	278 timmar eller 11.5 dagar
1 sextiondels sekund	16.667	4.6 timmar
En flugas vingslag (enl Isaac Asimov — se källfört)	3.160	53 minuter
Addera 5 och 7 i maskinspråk		
LDA 5	2	2
CLC	2	2
ADC 7	2	2
STA 251	3	3
Totalt	9 miljondels sekund	9 sekunder

fjärde och halv timma, skulle Du då anse Dig ha för tung arbetsbelastning? Svaret på detta är analogt med "interrupt"-(avbrotts-)rutinen som inträffar 60 gånger per sekund i VIC 20 och VIC 64. De 4,5 timmarna är tidsåtgången för att "köra" basprogrammet och halvtimmes rasten är "interrupt"-rutinen.

Hur lång tid tar det att köra ett program? Det beror på vad programmet gör. om man använder vår modifierade klocka, kan vi t ex beräkna en enkel addition med Basic. Additionen 5 + 7 tar 0,5 miljondels sekund. Det är 500 "tickningar" med vår modifierade klocka eller 500 sekunder på vår "uppförstorade" klocka. Titta nu tillbaka på tabellen på maskinspråksprogrammet, de adderade också fem och sju, det tog bara nio sekunder. Genom att dividera 500 med nio, ser vi att maskinsspråksprogrammet var 50 gånger snabbare. Vid en tillbakablick på BASIC, tog det 500 sekunder så är åtta minuter av 11,5 dagar fortfarande ganska bra. Faktum är att vi kan addera fem och sju ungefär 2.000 gånger på en miljondels sekund.

Nu har vi ödslat tillräckligt med tid på det här ämnet, så låt oss fortsätta. Ja, just det, angående frågan i början av texten.

Om Du väljer alternativ 3, har Du fel. Det genomsnittliga avståndet till månen är 420.464.000 yard. En "femcentare" per yard blir sammanlagt 21.023.200 dollar.

Gissade Du på alternativ 2 hade Du också fel. Avståndet mellan New York och Los Angeles är 177.027.840 inch. En "tio-centare" per inch blir sammanlagt 17.702.784 dollar.

Gissade Du på alternativ 1 hade Du rätt. Grattis!

Dagen då jag satte ihop den här frågeleken, var priset på guld 395 dollar per ounce. Två ton är 4.000 pounds (= 373 gram) eller 64.000 ounce. Vid 395 dollar per ounce blir det sammanlagt 25.280.000 dollar.

Storlek

Vi ska nu vända vår uppmärksamhet mot storleken av 6502-"chipen". De flesta människor har sett ett "chip"-hölje förut (som somliga kalla "caterpillar"). Den logiska slutsatsen av detta är att säga: "Så det där är ett 6502-"chip"." Nåja, det är inte helt rätt. Den svarta plast- eller keramiklådan med 28 kontakter ("ben") är faktiskt en behållare för 6502-"chipen". Själva chipet är en liten silverrektangel (en bit silicon) på 3 × 4 mm. Ungefär samma storlek som den lilla grafiska lådan på framsidan av Din VIC:s tangenter. (Om Du äger en PET/CBM är dess storlek ungefär som curson (markören) på bildskärmen.)

Men i alla fall, en liten silverrektangel gör inte att en microprocessor fungerar. Det måste finnas någonting på den, förstår Du! Visst finns det det. Faktum är att det finns över 5.000 "någonting" på den lilla rektangeln. Ett "någonting" kallas i elektroniken för komponent. Den kan vara en transistor, ett motstånd eller en förstärkare (det finns andra, men dessa är de vanligast förekommande).

Nu ska vi titta på komponenter som är så små att 5.000 stycket får plats på en enda "chip". Om 6502-"chipet" är 3 × 4 mm och vi delar det i 5.000 bitar, kommer varje bit att vara omkring 50 micrometer i kvadrat. Det är en 50 miljondels meter.

För att Du ska få en uppfattning om hur litet det är, ska vi jämföra det med storleken på ett sandkorn. Enligt Isaac Asimov är ett sandkorn omkring 200 micromillimeter stort. Vi skulle kunna placera 16 stycken av våra komponenter på ytan av ett sådant sandkorn. För "landkrabbor" kan det jämföras med att storleken på ett nålsöga är cirka 0,25 mm. (Enligt författaren: "Jag har en något skev uppfattning om hur man mäter det.") På den ytan får man plats med 100 komponenter.

Om vi nu ville förstora storleken på en komponent till något mer fattningsbart — en fotbollsplan till exempel. En fotbollsplan är ungefär 100 meter lång. Då skulle vi behöva förstora vår komponent två miljoner gånger. Hur stort är det lilla silver-"chipet" nu? Ja, det är ungefär två miljoner gånger större eller en yta på 3,5 × 5 miles (en mile = 1.609 m).

Vad är då storleken på höljet, eller "catepillaren"? Den är i normal fall cirka två inches lång och en halv inch bred. Men nu är den . . . 64,5 miles lång och 17 miles bred. Faktum är att om vi låter den stå på sina kontaker eller "ben" skulle den vara tio miles hög. Analogt med detta, hur stor skulle då VIC:en vara?

Ytan skulle vara 512 miles, 236 miles djup och 74 miles hög.

Trots att detta verkar vara fascinerande, ska man hålla i minnet att 6502:an inte är ett speciellt "tätt" "chip". Med andra ord, 5.000 komponenter på ett "chip" är inte speciellt mycket. Dagens banbrytande teknologi, med 256.000 bitars minnes-"chips", kan mer än en halv miljon komponenter finnas på ett enda "chip". För att återigen referera till ett nålsöga, skulle man på den ytan kunna placera omkring 2.500 komponenter.

Det här var alltså en liten resa med 6502:an i tid och rum. Jag är väl medveten om att den här sortens artiklar inte är späckade med fakta, men det kan göra detsamma. Det är nu hög tid att stanna upp och tänka efter vad Du egentligen arbetar med. Om Du förstår och uppskattar en dator, kan Du lära Dig använda den på rätt sätt. Använd den till fullo, på samma gång som Du lär Dig inse dess begränsningar.

Luleå 1984-02-10

Till VIC-Rapport
Redaktionen

Hej!

Jag har gjort ett program för rättning av lotto. Trots att programmet ser ganska enkelt ut, klarar det följande:
Du kan mata in upp till 400 spelfält, - enkla eller system med upp till maximalt tillåtna 11 kryss (=330 tips). Eftersom programmet är så enkelt fordras ingen rutin för datafil till band, då man helt enkelt bara sparar hela programmet med inmatade data-satser.
Programmet rättar sedan, efter att den korrekta lottoraden matats in, och skriver ut hur många rätt som finns på varje spelfält.

Beskrivning av programmet:

Rad 10: DIMensionering av matriser.
Rad 12: Kontroll om spelfält matats in (avläsning av data på rad 10000). Om ej inmatade data finns, går programmet automatiskt till rad 90, som uppmanar till inmatning av spelfält i form av data-satser, och skriver ut en modell för detta.
Rad 14-20: Ger möjlighet till inmatning av nya spelfält, via hopp till rad 95 och en listning av befintliga datasatser.
Rad 30-34: Läser av inmatade spelfält, och håller rätt på dels antalet spelfält, dels antalet inmatade nummer på varje spelfält. Vidare sker en kontroll av om man matat in fel, så att något spelfält har färre än 7 nummer eller fler än 11, - eller om något nummer är högre än 35. Vid sådana fel sker hopp till rad 96, som skriver ut vilket spelfält felet finns på, och automatiskt listar spelfälten.
Rad 35-38: Subrutin för avslutning av varje inläst spelfält.
Rad 40-46: Rapport av hur många spelfält av varje sort som lästs in.
Rad 50-56: Inmatning av rätt lottorad, med tilläggsnummer. Vid feltryckning uppmaning att korrigera det felinmatade talet.
Rad 60-78: Själva rättningsrutinen, och utskrift av antalet rätt, och antal tilläggsnummer, för varje spelfält. Resultatet presenteras 10 rader i taget, med automatisk fortsättning vid tryck av Return.
Rad 90 och framåt har beskrivits ovan.

Om antalet spelfält begränsas (DIM på rad 10), och utskriftsraderna redigeras om, bör programmet även fungera på VIC 20, då det ej innehåller några poke-kommandon eller andra konstigheter.

Med önskan om god lotto-tur!

Hälsningar!
Birger Gran
(Repslagarg.3A
951 35 Luleå)

```
5 REM * LOTTO-KONTROLL * AV BIRGER GRAN
10 DIMA(400,12),LR(12):J=1:I=1
12 READA:RESTORE:IFA=99THEN90
14 PRINT"{CLR}{DOWN}VILL DU MATA IN NYA SPELFALT? (J=JA)"
16 GETA$:IFA$=""THEN16
20 IFA$="J"THEN95
30 READ A(J,I):IFA(J,I)=99THENJ=J-1:GOTO40
32 IFA(J,I)=0THENGOSUB35
33 IFA(J,I)>35THEN96
34 I=I+1:GOTO30
35 I=I-1:IFI<7ORI>11THEN96
36 LR(I)=LR(I)+1
38 J=J+1:I=0:RETURN
40 PRINT"{CLR}{DOWN}DU HAR MATAT IN:{DOWN}"
42 FORI=7TO11:IFLR(I)=0THEN46
44 PRINTLR(I);" SPELFALT MED ";I;" NUMMER"
46 NEXT
50 PRINT"{DOWN}MATA NU IN RATT LOTTORAD -"
51 PRINT"{DOWN}7 ORDINARIE OCH 2 TILLAGGSNUMMER{DOWN}"
52 FORI=1TO9
54 INPUTR$:V=VAL(R$)
55 IFV<1ORV>35THENPRINT"FEL,TA OM SISTA!":GOTO54
56 R(I)=V:NEXT:S=0
60 FORK=1TOJ:R=0:T=0:FORL=1TO11:FORI=1TO7
62 IFA(K,L)=R(I)THENR=R+1
64 NEXTI:FORI=8TO9:IFA(K,L)=R(I)THENT=T+1
70 NEXTI,L:IFS=0THENPRINT"{CLR}{DOWN}SPELFALT","ORD.RATT","TILL.NR"
74 PRINTK,R,T:IFR>4THENPRINT"* * GRATULERAR! * *"
76 S=S+1:IFS=10THENS=0:INPUT"<TRYCK RETURN>";RE$
78 NEXTK:END
90 PRINT"{CLR}{DOWN}MATA IN DINA SPELFALT SOM DATASATSER"
91 PRINT"{DOWN}BORJA PA RAD 101."
92 PRINT"{DOWN}AVSLUTA VARJE SPELFALT MED ',0'{DOWN}"
94 PRINT"101 DATA":PRINT"102 DATA":PRINT"O.S.V.":END
95 PRINT"{DOWN}MATA IN DINA NYA SPELFALT EFTER DESSA:{DOWN}":GOTO99
96 PRINT"{DOWN}FEL INMATAT SPELFALT NR ";J
99 FORT=1TO2000:NEXT:LIST100-9999
10000 DATA99

READY.
```

Programmering för handikappade användare

Robin Nixon framställer en önskan om att mjukvara skräddarys för individuella användarebehov. Inte förrän detta finns tillgängligt kan påståendet att micros "hjälper de handikappade" rättfärdigas.

Robin Nixon är dator-bibliotekarie på Seven Springs Cheshire Home, Tunbridge Wells.

På sista tiden har det skett en märkbar ökning av antalet mikrodatorer som används av handikappade. Trots att teknologin har varit tillgänglig länge, användes den inte fullt ut förrän priserna kom ner till användarnas nivå.

Nu när marknaden översvämmas av mikrodatorer, har nya problem uppstått. Det första gäller input-anordningar. På grund av handikappens olika slag finns det ingen hårdvara som kan massproduceras billigt. Varje individ behöver indivuellt utformad utrustning. Det är inte ett så stort problem förrän man börjar skriva mjukvaran och då upptäcker alla möjligheter till input som programmet måste ta med i beräkningarna.

Det andra problemet är output. Många handikappade personer har synsvårigheter, så en TV-ruta är inte alltid bästa lösningen. Syntetiskt tal går framåt med stormsteg, men är fortfarande svårt att använda — speciellt för en blind person. Och tänk på de människor som är både blinda och döva! På grund av att alla dessa problem måste lösas individuellt finns det en markerad brist på mjukvara som är skriven speciellt för de handikappade.

Att skriva program för handikappade kan vara både mycket utmanande och också mycket tacksamt. När man ser någon som aldrig har kunnat skriva ett brev eller spela ett spel utan att lita till hjälp från någon annan, när du ser denna person använda din mjukvara och sålunda bli mer oberoende, då vet du att du har skapat något värdefullt. Trots att en översikt av hur man skriver för handikappade är till hjälp, finns det ingen verklig ersättning för att faktiskt besöka ett hem eller en skola och lära känna de inblandade människorna. Detta inkluderar personalen eftersom de har en större insikt än någon annan, undantaget den person det verkligen gäller, i förmågan hos varje person. Dessa är de människor som kommer att tala om exakt vad som behövs och ge goda råd om hur man skall gå tillväga.

De mest använda input-anordningarna är baserade på ett scan-system som normalt använder en eller två knappar. Vid användande av en knapp rör sig en cursor horisontalt längs med en alfanumerisk matris — se figur 1. För att skriva bokstaven y trycks första knappen ner och hålls nertryckt tills cursorn har flyttat sig till *bak* dvs "backspace" — se figur 2. När man släpper knappen avläser cursorn neråt. När den valda bokstaven lyses upp av cursorn trycks knappen ner igen och bokstaven läses då in i datorn — se figur 3.

Tvåknapps-moden är liknande med undantag för att den första knappen används för horisontell avläsning och den andra för vertikal avläsning. När den första knappen trycks ner en gång till läses bokstaven in. Matrisen kan ersättas när som helst av en kommateringspanel, en sifferpanel eller en ord-eller fraspanel.

Det kan tyckas vara en mycket långsam metod, men många handikappade användare har så begränsad rörelseförmåga att ett sådant här system är det bästa tills talregistreringsteknologin når en mycket högre standard. Naturligtvis kommer många människor att kunna använda fler än två knappar, inläsningshastigheten ökar för varje knapp.

Metoderna för input är praktiskt taget obegränsade; mycken tankeverksamhet måste läggas ner på urval och utveckling av input för varje individuell användare. Frånsett de två mest vanliga outputs, video och skrivare, finns det mycket få andra som är lätta att använda. Syntetiskt tal är kanske den mest använda av de andra alternativen. Talet är förståeligt och kommer att bli mycket bättre eftersom mycket arbete läggs ner på detta område. Så småningom kommer maskinvaran också bli billigare och lättare att använda.

Många handikappade användare har någon form av ögonspänningar eller andra synrubbningar. Så om en visuell display väljs måste den vara exceptionellt tydlig och så stor som möjligt. Ett sätt att göra video output tydligare är att använda två skärmar. Den första kunde innehålla bokstavsmatris och annan relevant information, t.ex. det sista ordet eller sista meningen som skrevs in. Storleken på bokstäverna på displayen skulle då kunna vara så stora som användaren behöver, och ändå, förhoppningsvis, lämna tillräckligt mycket utrymme på huvudskärmen för att köra programmet.

Den output som du tycker är bäst kan vara mycket förvånande. Jag känner en ung man som har extremt dålig syn och knappast någon hörsel. Han har en handburen summer, från vilken han känner vibrationerna från en morse-kod som datorn översätter från text som lagrats i dess minne. Han är en hängiven fotbollsanhängare och tycker om att följa resultaten. I stället för att stava dem för honom med fingerspråket i hans hand, så är det myc-

ket lättare att skriva in dem på datorn och låta honom gå över dem i sin egen takt.

Vilket program du än skriver är det väsentligt att komma ihåg att de flesta användarna är mycket oerfarna vad det gäller användande av datorer. En Hjälp-meny måste alltid finnas tillgänglig. Programmet måste också vara intressant och inte för komplicerat. En innehållsrik handbok är ett plus eftersom många handikappade användare måste lita till outbildad assistans.

Spelprogram kan vara mycket användbara i en miljö där det helt naturligt finns mycket fritid. De mest populära spelen är Scrabble (Alfapet), Monopol, Otello, Schack och Dam (eller Bräde). Snabba spel som t.ex. Space Invaders och Pacman skulle vara speciellt välkomna om en enkel inputmetod kunde tas fram och spelhastigheten kunde varieras med hänsyn till dem med långsam reaktionsförmåga.

Trots att det finns många kommunikationsprogram är de flesta baserade på samma avläsningssystem. Ett möjligt alternativ är att utveckla ett språk baserat på tankar och symboler, vilka var och en matas in i datorn och kombinationen översattes till svenska.

Bra program för att rita bilder och kurvor för matematiska funktioner är det ont om. Ett program som jag ännu inte har sett är en rutin för att tillåta användaren att läsa in och köra Basic program — inte så lätt som det låter.

Jag har med avsikt inte varit utförlig i min beskrivning av den programvara som nu finns tillgänglig. Det finns en god möjlighet att du, om du får tänka helt fritt, kan komma fram till helt nya tänkesätt vad gäller utförande av program för handikappade. Men om du behöver hjälp, skriv till författaren vid Seven Springs Cheshire Home, Penbury Road, Tunbridge Wells, Kent TN2 4NB där det finns ett nationellt programbibliotek för handikappade. Vi vill att dina program delas av så många människor som möjligt.



ÄVENTYR PÅ VIC-20/64



Kallsvettig och med skräcken i halsen, smyger du omkring längs kalla och fuktiga källargångar, på jakt efter den försvunna skatten. Du vet att ingen som försökt sig på bedriften att ta den tillbaka till civilationen har återvänt med livet i behåll.

Du ser lampan flimra till och vet att det är slut med dej om du inte lyckas hitta ingången till den hemliga grottan innan dina batterier är slut. Och plötsligt blir allt kolsvart, du känner kalla svettdroppar rinna efter nacken och paniken som kramar magen.

Du slår förtvivlat på lampan för att få liv i den igen, men den är död. Du rusar iväg i mörkret för att finna en plats att gömma dej på, så länge lampan lös så höll du monstren borta men nu kan allt hända.

Du snubblar till och faller raklång på stengolvet, kvickt vänder du om för att se vad du snubblat på, då hör du de hasande stegen bakom dej. Du rycker till och får tag i ett järnföremål, EN NYCKEL, ropar du till. Det är den! Nyckeln till grottan. I din lyckoberusning står du kvar för länge . . .

Vad är nu detta? och har detta med datorer att göra?

Jo, detta var en liten snutt ur ett ADVENTURESPEL. Många vet redan vad det innebär, men för dej som inte vet det, och för dej som vet men inte hur du själv kan göra dom, handlar den här artikelserien.

Den är menad att rikta sig till alla som i första hand har en VIC-20, och som är nybörjare. Men som sagt, att göra ett ADVENTURESPEL har sina sidor, så du som redan kan mycket kanske inte vet hur man gör ett sådant här spel.

Här hade jag tänkt att på ett enkelt sätt visa hur det kan gå till.

Att spela ett bra Adventurespel är (tycker iallafall jag) otroligt fängslande, dels därför att man blir bara mer och mer sporrad att ta reda på lösningen. Det hör nu inte till den enklaste sak att lyckas med det inte, det kan i vissa spel ta upp till flera månader. Det är som att vara detektiv, allt får du lista ut själv tillsammans med ''Din man på plats''.

Det enda du oftast får reda på när du börjar spela är att någon i spelet frågar dej . . .

— VAD SKA JAG GÖRA NU?

Sedan är det upp till dej att tala om vad som skall göras, och då har du som ända hjälp din egen fantasi samt de tillgångar spelet bjuder, i form av kommentarer ifrån ''Din utsände''. Vad han ser, hör och var han är.

Spelet bygger nästan bara på strängbehandling. Här hjälper ingen kunskap i supersnabba reaktioner med din väloljade joystick inte. Nej, bara din tankeverksamhet är väloljad så räcker det spelet igenom.

Hur som helst. Datorn förväntar sig alltså en mening ifrån dej, som t.ex. — TA LAMPAN, eller LÄS LAPPEN. När den fått en mening så behandlar datorn den på följande sätt. Först tar den reda på om det är ett eller två ord du skrivit, och det gör den genom att skilja orden åt vid mellanslaget. LÄS blir ORD 1 och LAPPEN blir ORD 2. Finns inget mellanslag blir allt du skrev ORD 1.

Det först ordet du skriver måste nästan alltid vara ett VERB, t.ex. TA, GÅ, ÖPPNA, SKJUT osv. Det andra ordet är antingen ett OBJEKT eller en riktning. Det vanligaste är att Adventures bara förstår två ord. Det går naturligtvis att få det att förstå hur många som helst, men det medför en desto krångligare programering, och längre. Jag kommer att till en början iallafall nöja mej med två.

Just den biten ser ut så här.

```
10 INPUT INS
20 FOR I = 1 TO LEN(INS)
30 IF MIDS(INS,I,1) = " " THEN 50
40 NEXT: 01S = INS: GOTO 70
50 01S = LEFTS(INS,I-1)
60 02S = RIGHTS(INS,LEN(INS)-I)
70 PRINT 01S: PRINTO2S
80 GOTO 10
```

Prova så får du se. Men mera om programering och förklaringar till den kommer senare.

Men nu för att överhuvud taget kunna göra ett eget ADVENTURE eller ÄVENTYRSSPEL som det heter på Svenska, så krävs det en historia eller ett mysterium kring vilket spelet ska kretsa och gå ut på. Och till nästa gång tänkte jag inte sätta dej vid datorn och knappa, utan vid skrivbänken. Och sätta dom grå på prov, du ska nämligen skriva en sådan historia. Den kan handla om vad som helst, bara den har ett antal bestämda platser eller rum som du kan gå till, samt innehålla diverse prylar vilka ska bidra till mysteriets lösning.

Det behöver nu till en början inte betyda att du skenar iväg till något hundrafemtirumsslott med lika många gångar, vrår och prår, häxor, tomtar och elaka monster i drösar, utan jag tycker att du kan vara nöjd om du får ihop mellan 5 och 10 olika platser, och kanske lika många saker.

Det är ganska mycket att tänka på redan här, och jag tänkte visa dej hur jag brukar gå tillväga.

Rita en karta över platsen för skådespelet, gör en liten lista över de olika platserna samt sakerna. Den kan se ut t.ex. så här.

Rumbeskrivning

1 Trapphus
2 Entré
3 Sovrum
4 Hall
5 Allrum
6 Vilorum
7 Kontor

Här finns . . .

1 En trappa
2 En dörr
3 En lapp
4 En nyckel
5 En dåligt gömd dörr
6 En nyckelknippa

Allt det här kan självklart bytas ut mot sträckbänkar, piskor och dylika tingestar för dom som nu önskar det. Men vad det nu än vara må, så är principen alltid den samma.

Vidare så är det lämpligast att du går i väderstreckens riktningar, alltså nord, syd, väst och öst. Därför att GÅ BAKÅT eller GÅ HÖGER, blir bara bökigt. Det allra enklaste är också om man kortar ner alla riktningar man ska kunna gå åt, till bara en bokstav. Så ska du t.ex. här ovan från rum 5 till rum 6, så skriver du bara Ö. Sedan kan du även skriva ner alla kommentarer du vill ska förekomma i ditt spel, t.ex.

Datorn frågar.

— VAD SKA JAG GÖRA NU?

Du svarar.

— TA NYCKELN.

Datorn säger.

— DET FINNS INGEN HÄR!

Osv. Detta för att du ska få en så överskådlig bild av spelet som möjligt, innan du börjar göra program av det.

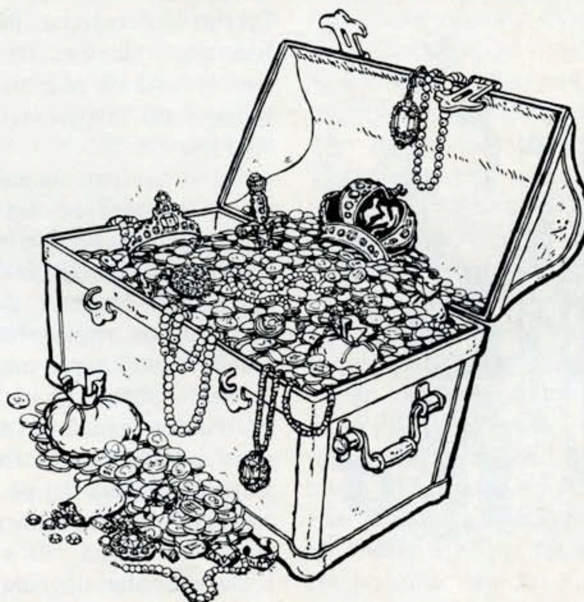
Det låter kanske både mycket och kanske krångligt, men när du kommit igång så flyter det både lätt och oftast alldeles för långt. Tänk bara som en dator när du skriver, INMATNING, BEARBETNING och UTSKRIFT. Men du måste ha med dessa kommentarer ifrån ''din man på plats'', därför att du dels måste vara i samma rum som de saker du vill ta, eller dörrar du vill öppna. Och du kan bara gå i de riktningar som rummen eller platserna tillåter.

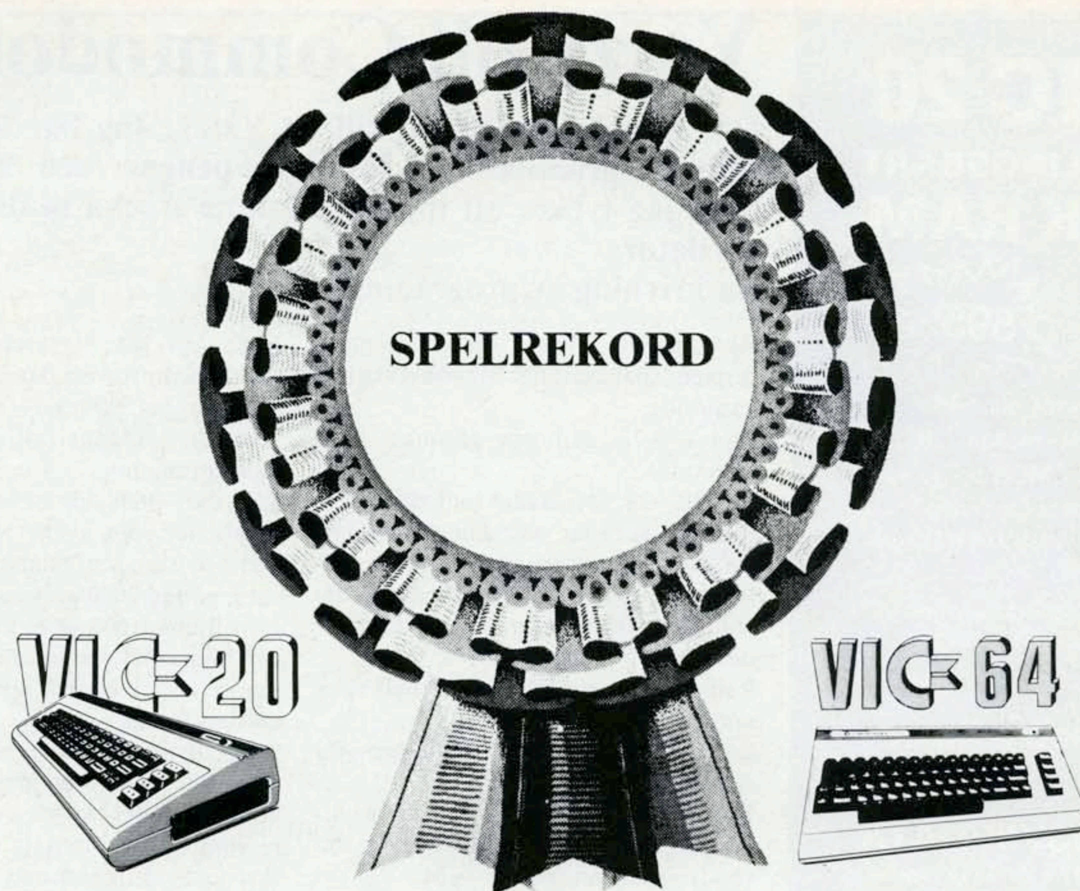
Och grötar det ihop sig för dej så får du mera hjälp i kommande nummer.

Jag hoppas att jag förklarat mej på ett förståeligt sätt och att det låter intressant. Så vässa pennorna och spar oljan till cellerna, så kommer du nog alltid på något.

Lycka till, till nästa gång!

Börje Törnby





JELLY MONSTERS VIC-1905

1.	1 631 860	Pelle Gustavsson, SUNDBYBERG
2.	763 650	Peter Johansson, LYCKEBY
3.	705 980	Anders Hellman, NOL

RAT RACE VIC-1909

1.	83 312	Per Bergström, VÄSTERVIK
2.	82 180	Magnus Eriksson, LJUNGBY
3.	70 200	Leif Lindström, ÖREBRO

SUPERLANDER VIC-1907

1.	88 000	Fredrik Gustafzon, GNESTA
2.	86 000	Thomas Johansson, YSTAD
3.	77 700	Tomas Lundh, SKÖVDE

PANIC 64 (Interceptor)

1.	5 300	Torbjörn Almén, SJÖBO
2.	4 200	Charlie Johansson, HANDEN
3.	2 100	Dr Darnio

GRIDRUNNER (LLamasoft)

1.	94 180	Niklas Lindberg, VÄSTERVIK
2.	58 610	Stuffe
3.	9 810	Maria Weijdling, PERS GÄRDE

JUMPMAN (EPYX)

1.	27 400	J. Aspengren, LINDOME
2.	17 400	Johan Bergqvist, TROLLHÄTTAN
3.	15 800	Joakim Ramsberg, MÖLNDAL

SPELREKORD

På grund av omorganisation av redaktionen har inte "SPELREKORD" kommit igång ordentligt. Därför är det ingen större förändring från det förra numret.

Flickor!

Skicka in fler rekord så startar snart en tjejliga.

Vänliga Hälsningar
SPELREDAKTÖRN
C.E.J.

Datum

Namn ....

Adress ....

Postnr, Postadress ....

Spel ....

Erhållna poäng ....

Intygas av målsman eller annan myndig person

Yatzy till Commodore 64

Här bifogas en listning till ett Yatzy. Jag har sett att det säljs programkasetter för dyra pengar med Yatzy-spel. Men jag tycker att man bör kunna trycka in det själv på sin dator.

Beskrivning av programmet:

Rad 20—30: skapar en 'ner-sträng' resp en 'space-sträng' för enkel skärmhantering.

Rad 40—70: namnger de olika nummer-valen.

Rad 72—78: förbereder ljudregistret.

Rad 79: anropar subrutin på raderna 3000—3040 som ger en fanfar, när spelet startar. Den anropas även när någon spelare får Yatzy eller uppnår bonuspoäng.

Rad 80—90: startar upp aktuell spelomgång.

Rad 100: här börjar huvudloopen för själva spelet.

Rad 140—150: ritar ut 5 'tomma tärningar'. För de vertikala strecken används Commodore +M, då det vänstra strecket annars raderas när siffran skrivs ut i tärningen.

Rad 160: 'Slår' tärningarna, och låter siffrorna 'snurra' inne i de fem tärningarna.

Rad 170: anropar subrutin 1000, som kollar om slaget blivit 'Yatzy' därefter skrivs kvarvarande kombinationer ut. (Rad 170—185).

Rad 190: anropar subrutin på rad 1100, som raderar 3 hela rader.

Rad 190—210: möjliggör val av antal tärningar som slås om. Här liksom i alla följande inmatningar sker dessa med strängar, som sedan omvandlas till tal. Detta för att slippa att programmet bryts av ett 'Error redo from start' vid feltryckning.

Rad 220—270: kollar vilka tärningar som ska slås om, och gör detta. Sedan görs en ny kontroll om 'Yatzy' inträffat. Flaggan 'K' avläses, för att avgöra om ytterligare ett omslag får göras. Är K = 1 hopp till 300, annars återgång till 190 för ytterligare ett omslag.

Rad 300: summerar tärningarna som hjälp för de kombinationer där hela summan får användas.

Rad 310—330: val av kombination för att spara slaget, koll om den kombinationen redan tagits, i så fall återgång för nytt val.

Rad 340—345: inmatning av aktuell poäng. Kontroll att denna ej överstiger tärningarnas summa (annat än vid 'Yatzy'). I sådant fall utskrift av en mild förmaning.

Rad 350—360: summering av de kombinationer som avgör rätt till bonus. Om bonusnivå uppnåtts anropas subrutin på rad 2000 som automatiskt lägger till bonuspoängen, avger en fanfar och hissar 'bonusflaggan' ,BO(J), som markerar att aktuell spelare tillgodoräknats bonus.

Rad 370: nästa spelare, nästa varv.

Rad 400—430: summerar resterande kombinationer, kollar av vilken spelare som uppnått högsta poängen, och skriver ut deltagarlistan med angivande av varje spelares poängsumma och en speciell markering av vinnaren.

Rad 440—460: ger möjlighet att börja om för en ny omgång.

Rad 1000—1020: subrutin för koll om man slagit 'Yatzy'. I sådant fall anropas subrutin på:

Rad 1200—1220: som blinkar med skärmen, skriver ut 'Yatzy' och därefter anropas fanfaren i subrutin 3000—3040.

Programmet är skrivet kompakt för att lättare kunna publiceras, men jag hoppas att dessa förklaringar underlättar att följa det.

Programmet är ju relativt enkelt och borde gå att köra även på VIC 20, om 'ner-strängarna' och 'space-strängen' anpassas till VIC 20:s radlängd och radantal, och om uttabuleringen av de fem tärningarna (rad 140—160 och 250) görs så tätt som möjligt. Givetvis måste även ljudkommandona göras om (rad 72—78 och 3000—3040). Poke-kommendona på rad 10 sätter skärm och ram-färg och ersätts med motsvarande för VIC 20.

Med detta tillönskas alla 'Yatzy-fans' mycket nöje!

BIR

```
5 REM * YATZY * FöR COMMODORE 64 AV BIRGER GRAN
10 POKE53280,5:POKE53281,0:PRINT"□":DIMK$(15),K(10,15),NE$(24)
20 NE$(0)="■":FORI=1TO24:NE$(I)=NE$(I-1)+"▣":NEXT:T$="          "
30 FORI=1TO39:SP$=SP$+" ":NEXT:FORI=1TO14:ST$=ST$+"* ":NEXT
40 K$(1)=" ETTOR"+T$:K$(2)=" TVAOR"+T$:K$(3)=" TREOR"+T$
50 K$(5)=" FEMMOR"+T$:K$(6)=" SEXOR"+T$:K$(7)=" ETT PAR"+T$
60 K$(9)=" 3-TAL"+T$:K$(10)="4-TAL"+T$:K$(11)="LITEN STEGE"
65 K$(4)=" FYROR"+T$:K$(8)=" 2 PAR"+T$:K$(12)="STOR STEGE"
70 K$(13)="KAK   "+T$:K$(14)="CHANS"+T$:K$(15)="YATZY"+T$
72 BE=54272:FORCL=BETOBE+24:POKECL,0:NEXT
74 POKEBE+5,85:POKEBE+6,85:POKEBE+12,85:POKEBE+13,85
76 DATA 25,30,18,209,33,135,25,30,42,62,31,165,50
78 DATA 60,37,162,42,62,31,165,50,60,37,162
79 GOSUB3000
80 PRINT"□▣"SPC(11)"◣* * Y A T Z Y * * ▣▣□":PRINT"HUR MANGA SPELARE";
90 INPUTM:FORJ=1TOM:SU(J)=0:BO(J)=0:FORK=1TO15:K(J,K)=0:NEXTK,J:MX=0
100 FORG=1TO15:FORJ=1TOM
110 PRINT"□▣"SPC(11)"◣* * Y A T Z Y * *":PRINTNE$(7);"□ ■SPELARE ";J;"■";
120 PRINT" TRYCK <SPACE>!◣"
130 GETA$:IFA$<>" "THEN130
140 PRINTNE$(7);SP$:FORT=1TO5:PRINTNE$(3)SPC(6*T)" ┌──┐"
150 PRINTSPC(6*T)" |  |":PRINTSPC(6*T)" └──┘":NEXT
160 FORT=1TO5:FORS=1TO20:X=INT(RND(1)*6)+1:PRINTNE$(4)SPC(6*T+1)X:NEXT
170 T(T)=X:NEXT:SU=0:K=0:GOSUB1000:PRINTNE$(7)" ■SPELARE ";J;"■DU HAR KVAR: "
180 FORK=1TO15:IFK(J,K)=0THENPRINTK;K$(K),
185 NEXT
190 GOSUB1100:PRINT"□HUR MANGA TARNINGAR VILL DU SLA OM"
200 OM$="":INPUT"(0-5)";OM$:IFOM$=""OR(VAL(OM$))>5THENGOSUB1100:GOTO200
210 OM=VAL(OM$):IFOM=0THEN300
220 FORO=1TOOM
230 U$="":INPUT"VILKA TARNINGAR";U$
235 IF(VAL(U$))=0OR(VAL(U$))>5THENGOSUB1100:GOTO230
240 U(O)=VAL(U$):PRINT"□";:NEXT:PRINT"◣"
250 FORO=1TOOM:FORS=1TO20:X=INT(RND(1)*6)+1:PRINTNE$(4)SPC(U(O)*6+1)X
260 NEXTS:T(U(O))=X:NEXTO:GOSUB1000:IFK=1THEN300
270 K=1:GOTO190
300 FORZ=1TO5:SU=SU+T(Z):NEXT:GOSUB1100:PRINT"□TARNINGARNAS SUMMA: ";SU
310 PRINT"HUR VILL DU SPARA SLAGET <NR>";:L$=""
320 INPUTL$:L=VAL(L$):IFL=0ORL>15THENGOSUB1100:GOTO310
330 IFK(J,L)<>0THENGOSUB1100:PRINT"REDAN TAGET":GOTO310
340 P$="":PRINT"VILKEN POANG";:INPUTP$:IFP$=""THENPRINT"□";:GOTO340
345 IFYA<5ANDVAL(P$)>SUTHENGOSUB1100:PRINT"AJA BAJA! INTE FUSKA!":GOTO340
350 K(J,L)=VAL(P$):IFL<7THENSU(J)=SU(J)+K(J,L)
360 IFBO(J)=0ANDSU(J)>63THENGOSUB2000
370 NEXTJ,G
400 FORJ=1TOM:FORL=7TO15:SU(J)=SU(J)+K(J,L):NEXTL:IFSU(J)>MXTHENMX=SU(J)
410 SP=J:NEXTJ:GOSUB1100:PRINTNE$(7)SP$:PRINT"RESULTAT:"
420 FORJ=1TOM:IFJ=SPTHENPRINT"■VINNARE:■";
430 PRINT"SPELARE NR ";J;",POANG:";SU(J):NEXT
440 PRINTNE$(20)"EN OMGANG TILL? TRYCK <*>"
450 GETA$:IFA$<>"*"THEN450
460 GOTO79
1000 YA=1:FORI=1TO4:IFT(1)=T(I+1)THENYA=YA+1
1010 NEXT:IFYA=5THENGOSUB1100:GOSUB1200:GOSUB3000:FORT=1TO2000:NEXT
1020 RETURN
1100 PRINTNE$(17);SP$:PRINTSP$:PRINTSP$:PRINT"□□□";:RETURN
1200 FORF=1TO3:POKE53281,1:FORV=1TO99:NEXT:POKE53281,0:FORV=1TO99
1210 NEXTV,F:PRINTSPC(4)"■";ST$:PRINTSPC(4)"* * YATZY!  GRATULERAR! * *"
1220 PRINTSPC(4)ST$;"□":RETURN
2000 SU(J)=SU(J)+50:BO(J)=1:GOSUB1100
2010 PRINTSPC(4)"■";ST$:PRINTSPC(4)"* * *   B O N U S !   * * *"
2020 PRINTSPC(4)ST$;"□":GOSUB3000:FORT=1TO1000:NEXT:RETURN
3000 POKEBE+24,15:POKEBE+4,33:POKEBE+11,17:FORXY=1TO6
3010 READH1,L1,H2,L2:POKEBE+1,H1:POKEBE,L1:POKEBE+8,H2:POKEBE+7,L2
3020 IFH1=50THENFORT=1TO150:NEXT
3030 FORT=1TO75:NEXTT,XY
3040 POKEBE+4,32:POKEBE+11,16:POKEBE+24,0:RESTORE:RETURN

READY.
```

Om FORTH — *Ett annorlunda språk*

av Ola Johansson

Som du säkert vet finns det flera språk än BASIC till VIC-datorerna. Man har till exempel kunnat läsa om språket COMAL här i VIC-rapport. Ett annat språk som är intressant är FORTH. Det är ett strukturerat språk, som betyder att man skriver programmet i överskådliga delar som tillsammans löser det problem man har. Det skiljer sig mycket från BASIC men för en datorintresserad är det alltid kul att lära sig något nytt!

Varför ska man då lära sig ett nytt språk? Räcker det inte med BASIC? I vissa fall kan det göra det, men ibland kan man behöva ett nytt verktyg till sina uppgifter. Det är ju som ett verktyg man använder programmeringsspråket, och som sådant kan det ha både för- och nackdelar. En av de nackdelar som BASIC har men inte FORTH är att det är svårt att skriva sitt program strukturerat (se inledningen). I FORTH är man ''tvungen'' att göra detta och om man skriver ett stort program är det nödvändigt om man vill kunna förbättra programmet senare (många har väl sett ett s k ''spagettiprogram'' med så många GOTO och GOSUB att man blir alldeles snurrig?)

En historisk tillbakablick

Efter ovanstående lilla beskrivning av en del skillnader mellan BASIC och FORTH är det dags att reda ut vad FORTH egentligen är och hur man programmerar i detta förträffliga språk. Själva idén till FORTH dök upp i huvudet hos en programmerare vid namn Charles H. Moore i början av 1970-talet. Han hade tidigare försökt förbättra de metoder han använde som programmerare, men under åren 1968—70 började språket FORTH ta form. Första gången FORTH användes i ''verkliga livet'' var 1971, då det användes i ett astronomiskt observatorium i USA. Under användandet på observatoriet utvecklades det ytterligare och 1973 bildades Forth Inc. av några entusiaster. Bland dem fanns förstås Charles Moore. 1976 kom en version som kallades microFORTH och samma år bildades i Europa gruppen EFUG (European FORTH Users Group). Några år senare bildades FIG (FORTH Interest Group), som har bidragit till den variant av FORTH som VIC-arnas FORTH bygger på: FORTH-79. Detta är en standardversion som har skapats för att inte FORTH ska råka ut för samma missöde som BASIC, nämligen det att det finns minst lika många olika BASIC-dialekter som det finns BASIC-datorer.

Vad kännetecknar FORTH?

FORTH skiljer sig mycket från andra språk. För det första består språket av en hel massa ord (ungefär som kommandon eller funktioner) ordnade i en sk ordlista, som man själv bygger ut under progrmmeringens gång, och för det andra så utförs alla beräkningar på en stack, som förklaras närmare i nästa stycke. Dessutom är FORTH interaktivt, vilket betyder att det hela tiden ''händer'' något när man arbetar med datorn. Du kan ju t ex jämföra med BASIC-ens kommandon LIST och RUN. I FORTH fungerar alla ord som en sorts kommandon. När du programmerar i FORTH skriver du olika definitioner (förklaringar hur något ska utföras) av ord, som du sedan kan kombinera till andra ord till du har ett eller ett fåtal ord som gör hela jobbet. De ord du själv skriver hamnar den ordlista som FORTH består av (se ovan). För att göra en försiktig jämförelse med BASIC igen (vilket är svårt, eftersom de har så lite gemensamt) så liknar det sättet med vilket du skriver program uppdelat i små subrutiner, programdelar, som gör varsin del av jobbet.

Stacken

Stacken är det ställe där alla tal som behövs i olika beräkningar hamnar. Där läggs också resultaten av beräkningarna, samt de tal som en funktion kan tänkas ge ifrån sig. Stacken fungerar enligt något som kallas postfixnotation, omvänd polsk notation eller RPN (kärt barn har många namn). De som känner till hur Hewlett-Packards miniräknare fungerar vet vad jag menar. Stacken kan tänkas fungera som en hög (''stack'' på engelska) av tal, ur vilken man bara kan plocka det översta. Dessutom måste alla tal som man vill räkna med finnas på stacken innan man utför beräkningen. Det ger som följd att man räknar ''baklänges'' i FORTH, alltså att man först skriver de tal man t ex vill lägga ihop och sedan skriver + —tecknet. Detta kan bäst beskrivas men några exempel, som finns senare i artikeln. Det kan kanske verka avigt i början, men efter en tid kan man inte klara sig utan stacken och RPN. För datorprogrammering är det dessutom utmärkt, eftersom (nästan) alla datorer räknar med en stack internt. Det gör t ex alla datorer som bygger på 6502-processorn och det gör bl a VIC-datorerna.

Finns nackdelar hos FORTH?

Javisst, det vore fel att säga annat. ALLA språk har fler eller färre nackdelar, även om en del påstår annat. En nackdel som man lägger märke till hos FORTH är att det inte finns decimaltal, FORTH räknar bara med heltal. För en del kan detta verka vara en allvarlig brist, men ofta klarar man sig med heltal, och om man inte gör det kan man faktiskt fixa det själv, med en smula programmering. I en del versioner av FORTH finns det t om inbyggt från början, dock inte i VIC-arnas FORTH. En annan nackdel, som främst är ekonomisk, är att man nästan måste ha en diskdrive, eller diskettstation som det också kallas, som är mycket dyr (3 495:— till VIC-arna enl prislistan 83-08-15) jämfört med en bandspelare. Nu går det faktiskt att använda bandspelare också, men det blir då ganska trögjobbat och inte det nöje som det borde vara.

En annan sak som BASIC- användare kan tänkas sakna är GOTO-instruktionen, som tillåter att man hoppar lite hur som helst i ett program. Denna instruktion kan dock uppmuntra till röriga program som är svåra att tyda, så saknaden bör bli kort. När vi är inne på ''GOTO- frågan'' vill jag citera Leo Brodie som i sin bok ''Starting FORTH'' skriver: ''By the end of this book you'll be telling your GOTO where to GOTO.'' Jag undrar vart . . .?

Smakprov

Nu kan det vara dags för några smakprov på hur det kan se ut när man använder FORTH. Exemplen är så gott som självförklarande, det som du eller jag skriver har jag strukit under. Efter detta kommer datorns svar. Varje gång datorn har utfört sin uppgift skriver den OK, för att tala om att den är klar för en ny uppgift.

3 5+. 8 OK (Punkten tar översta talet från stacken, resultatet, och skriver ut det på skärmen.)

3 5—. —2 OK

.''FOO BAR'' FOO BAR OK
(ordet .'' skriver ut text på skärmen.)

Nedan finns en lite mer avancerad beräkning, därför visar jag till höger om exemplet hur stacken ser ut efter varje rad. Det ''översta'' talet i stacken ligger längst till vänster.

3 5 OK	5 3 — — — — —
+ OK	8 — — — — — —
17 13 OK	13 17 8 — — — —
— OK	4 8 — — — — —
/ OK	2 — — — — — —
2 OK	— — — — — — —

Eller om man skriver det på en rad:
3 5 + 17 13—/.2 OK _ _ _ _

För att tydligare förklara vad som har utförts, skriver jag räkningen en gång till, men på det sätt man brukar skriva i matematiken:

$$\frac{3+5}{17-13}=2$$

För att ge några ytterligare exempel på hur det kan se ut när man använder FORTH så måste jag först förklara hur man skriver program, definierar ord, i FORTH. Det är egentligen ganska enkelt. Om du t ex vill ha ett ord som skriver HEJ DU GLADE varje gång du utför ordet, som du kan kalla HEJ, så skriver du:

: HEJ .'' HEJ DU GLADE '' ; OK

Nu räcker det med att skriva HEJ så utförs ditt nya ord, precis som om det alltid skulle ha funnits hos FORTH.

HEJ HEJ DU GLADE OK

En definition börjar alltid med ordet : (kolon), som vill ha namnet på den nya funktionen och sedan de ord som talar om hur ordet ska fungera. Det hela avslutas med ; (semikolon). Nu ytterligare några exempel:

: ÖKA-MED-TVÅ 2 + ; OK
11 ÖKA-MED-TVÅ . 13 OK
: + FYRA ÖKA-MED-TVÅ ÖKA-MED-TVÅ ;OK
38 + FYA . 42 OK
17 + FYRA + FYRA. 25 OK
: PRINT .; OK
17 + FYRA + FYRA PRINT 25 OK

Som du kan se på exemplen ovan så kan du ändra och bygga ut FORTH precis som du vill. Du kan t ex ta ordet ÖKA-MED-TVÅ och använda det i ordet + FYRA, eller ''byta namn'' på . (punkt) till PRINT om du vill.

Detta gör FORTH till ett väldigt flexibelt språk, som hela tiden anpassar sig till den som använder datorn. För att jämföra FORTH med andra språk kan jag nämna en tabell som jag såg i tidningen Mikrodatorn nr 9-83, där författaren hade jämfört olika språk med varandra.

Hur är FORTH jämfört med andra språk?

De språk som fanns upptagna i ovan nämnda tabell var bl a BASIC och FORTH, och de hade fått olika ''betyg'' efter deras resp förmåga att vara bra för nybörjare, snabbhet, allsidighet osv. Bland de saker som FORTH hade fått höga betyg i fanns allsidighet. Detta kunde du ju se redan i exemplen ovan, men vänta tills du lärt dig mer . . . (för du har väl blivit nyfiken på FORTH, eller . . .). Även i snabbhet och minnesbehov, d v s hur mycket programminne som behövs hade FORTH fått höga betyg. I dessa tre olika värderingar hade däremot BASIC fått medel eller lägre, men desto högre angående lätthet att lära och hur lätt det ä att få tag på (det finns ju i nästa alla små- och mikradatorer). Om man sedan tittar på allmänna bedömningen så hade FORTH högsta (!) medelbedömning, vilket kan vara intressant att veta (BASIC kom på andra plats av fem språk). Det enda som FORTH hade fått dåligt på var i hur lätt det är att lära. Detta kan nog stämma, men kanske bara hos de som redan kan t ex BASIC. FORTH är nämligen helt annorlunda jämfört med andra språk. Själv kunde jag både BASIC och del andra spåk innan FORTH, så jag fick verkligen försöka glömma det jag redan kunde om BASIC och se på FORTH med nybörjarens ögon. De som nu redan kan ett annat språk ska inte stick huvudet under jorden bara för det. Man har ju alltid nytta av att kunna *programmera*, samt att känna till en del om datorer. Boken ''Starting FORTH'' av Leo Bodie tar fasta på detta och har skrivit boken för två typer av människor. Dels de som redan kan programmera datorer i ett annat språk, och dels de som inte kan programmera alls. Den boken rekommenderar jag starkt till de som vill läsa mer om FORTH.

I en senare artikel kommer jag att lära ut FORTH till den som vill få en

djupare inblick i språket, samt ge en del tips till de som redan har köpt eller funderar på att köpa FORTH till sin VIC. Denna tillsats kostar 495:— och detta pris är också ur prislistan från 83-08-15.

Om ni har några frågor om FORTH, allmän eller speciellt för VIC-FORTH, så kan ni skicka in dem till tidningen så ska jag försöka svara på en del av dem i mån av kunskap och tid.

Sammanfattning

Slutligen vill jag göra en liten sammanfattning av en del viktigt ur artikeln:

- FORTH är ett annorlunda språk.
- Det skapades på 70-talet av Charles H. Moore.
- FORTH består av en massa ord, och kan definiera egna sådana. Då man gör det kan man säga att man bygger ut språket.
- Alla beräkningar sker på en stack i omvänd polsk notation, RPN.
- FORTH har ofta inte decimaltal från början, inte VIC-FORTH t ex, men om man inte klarar sig utan kan man fixa det själv.
- Man bör helst ha en diskettstation.
- Boken ''Starting FORTH'' av Leo Brodie är en bra bok.
- FORTH är ett trevligt och intressant språk, som är kul att lära sig och att använda.

Några lästips:

- Starting FORTH, av Leo Brodie.
- Tidningen BYTe, augustinumret 1980. Har FORTH som tema.
- PET-FORTH handboken. Bra att ha om man har köpt VIC-FORTH.

KALENDERPROGRAM

Almanackor är ju oumbärliga hjälpmedel för att orientera sig i tillvaron. Felet med dem är att de bara gäller för ett år i taget. Den som vill ha en evighetskalender kan faktiskt programmera in en sådan på sin VIC, fast det blir förstås ett ganska vidlyftigt program.

Problemet är för det första att vi bytte ''stil'' år 1753, då vi fick den gregorianska kalendern i stället för den julianska (som alltjämnt används i Sovjet och en del andra länder). På så sätt infördes en ny skottårsräkning bl a. Det är ju skottåren som ställer till trassel i ordningsföljden, bl a när man ska räkna ut veckodag för ett visst datum.

Det finns olika sätt att lösa skottårsproblemet i ett kalenderprogram. Ett är att utgå från regeln att samma veckodag återkommer på samma datum vart 28:de år. Det stämmer så länge man bara håller sig inom ett århundrade. Överskrider man sekelgränserna så blir det fel.

Det program som presenteras här är i princip mycket enkelt. Det baserar sig nämligen på fenomenet, att året inte har jämnt 365 dagar utan faktiskt omspänner 365,2425 dygn. Avkortat till 365,25 fås automatiskt ett tillägg för skottdagen vart fjärde år utan några krångliga IF-satser. De enda undantagen gäller sekelskiftena 1800 och 1900, som inte är skottår. Där hoppas skottårsmånaden februari 29 dagar över med en IF-sats.

Programmet, som täcker alla år mellan 1753 och 2100 är ganska långsamt, eftersom alla räkningar sker från 1753. Men det är i stället så mycket säkrare.

När infaller påsken

Det andra programmet räknar ut på vilket datum påskdagen infaller under valfritt år. Det är ju som bekant lite olika med den saken. Det var på kyrkomötet i Nicea år 325 som det bestämdes att påsken skulle inträffa *första söndagen efter första fullmånen efter vårdagjämningen, dock inte senare än den 25 april.*

Det där kan ju låta lite avskräckande för en programmerare, men problemet är till största delen redan löst av den geniale tyske matematikern Karl Friedrich Gauss, som år 1800 konstruerade en formel för uträkning av påskterminen.

Den här formeln är ganska inveckIad. Det tar tid att hanskas med den för hand med block och penna, och med fickräknare går det inte mycket lättare. Däremot är den tämligen enkel att programmera på datorn.

Formeln lyder:
Önskat årtal divideras i tur och ordning med talen 19, 4 och 7. När det är gjort tar man *resterna* efter divisionerna och kallar dem förslagsvis A, B och C. Till 19*A lägger man ett tal, som vi kallar M och som är olika beroende på århundrade. Ligger årtalet före 1800, så är M = 15. För åren 1800—1900 har M värdet 23 och för 1900 fram till 2099 är M = 24.

19A + M divideras med 30 och *resten* kallar vi D. Sen tar vi 2*B + 4*C + 6*D och lägger till ett tal N, som skiftar värde precis som M. Före 1800 är N = 6, 1890—1899 är N = 4 och mellan 1900 och 2099 har N värdet 5.

2*B + 4*C + 6*D + N ska sen divideras med 7. *Resten* kallar vi E. För att få påskdagens datum adderar man till sist D + E + 22. Om talet man får är = 31 infaller påsken på givet datum i mars. Är talet 31 minskar man 31 och får datum i april. Skulle det mot förmodan inträffa att svaret blir den 26 april, vilket det följande programmet inte är garderat mot, så infaller påksdagen i verkligheten en vecka tidigare.

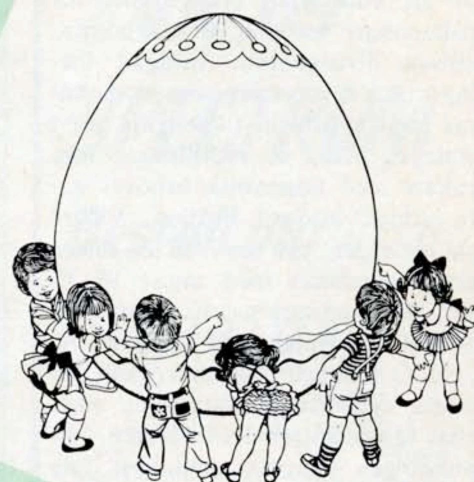
Conny Jonsson

Vilken veckodag?

TVÅ KALENDERPROGRAM av Conny Jonsson

VILKEN VECKODAG?

```
1 PRINT"J"
5 PRINT"*****VILKEN VECKODAG?*****"
10 INPUT "VILKET ÅR";A
11 INPUT "VILKEN MÅNAD";B
12 INPUT "VILKET DATUM";C
30 FOR I=1753 TO A
40 D=A-1753
50 S=D*365.25
60 IF D>47 THEN S=S-1
70 IF D>147 THEN S=S-1
75 S=INT(S)
80 NEXT I
90 FOR J=1 TO 12
100 READ D
110 IF J=B THEN 132
120 S=S+D
130 NEXT J
132 IF A=1800 OR A=1900 THEN 145
135 IF J=2 AND A/4=INT(A/4)THEN D=29
140 IF J>2 AND A/4=INT(A/4) THEN S=S+1
145 GOTO 150
150 DATA 31,28,31,30,31,30,31,31,30,31,30,31
152 PRINT:PRINT
155 PRINT"DEN";C;"/";B;"ÅR";A;"ÄR EN ";
160 S=S+C
165 IF C>D THENPRINT"LÄR DIG MÅNADERNA!"
170 S1=S-INT(S/7-0.1)*7
180 ON S1 GOSUB 201,202,203,204,205,206,207
190 END
201 PRINT"MÅNDAG":RETURN
202 PRINT"TISDAG":RETURN
203 PRINT"ONSDAG":RETURN
204 PRINT"TORSDAG":RETURN
205 PRINT"FREDAG":RETURN
206 PRINT"LØRDAG":RETURN
207 PRINT"SØNDAG":RETURN
```



När infaller påsken?

```
1 PRINT"J"
5 PRINT"NÄR INFALLER PÅSKEN?"
10 INPUT"VILKET ÅR GÄLLER DET";S
20 A=S-INT(S/19)*19
30 B=S-INT(S/4)*4
40 C=S-INT(S/7)*7
50 IF S<1800 THEN M=15
55 IF S<1800 THEN N=6
60 IF S<1900 THEN M=23
65 IF S<1900 THEN N=4
70 IF S=>1900 THEN M=24
75 IF S=>1900 THEN N=5
80 D=19*A+M-INT((19*A+M)/30)*30
90 E=2*B+4*C+6*D+N-INT((2*B+4*C+6*D+N)/7)*7
100 P=D+E+22
110 IF P>31 THEN GOTO 130
120 PRINT"ÅR ";S;"INFALLER PÅSKEN DEN ";P;"MARS"
125 GOTO 5
130 PA=P-31
140 PRINT"ÅR ";S;"INFALLER PÅSKEN DEN ";PA;"APRIL"
145 GOTO 5
150 END
```

Konvertering av stryktipsprogrammet ''DATA-TIPS''



I föregående nummer av VIC-RAPPORT, 5/6 1983, presenterades stryktipsprogrammet ''DATA-TIPS''. Programmet skrevs ursprungligen för tipsmatcher med lag ur de båda högsta, engelska divisionerna. Bland de förfrågningar som ställts till programmakaren märks frågan om hur att konvertera programmet för tipskuponger med lag ur de högsta, svenska divisionerna. Bifogad listning visar vilka rader som skall ändras samt eventuellt tillskrivas. Sist i listningen visas de radtillsatser som brukare med floppydisk behöver ändra utöver bifogad listning. Vidare skall tilläggas, att din VIC-20 måste vara expanderad med minst 16 K-Ram. Programmet passar även VIC-64. VIC-apparater med tecknen, Å, Ä och Ö som extra tillbehör, skall observera den programsats som medföljer utbytessatsen från Handic. Filhanteringen fungerar nämligen inte förrän rekommenderad programsnutt lästs in och exekverats.

Läs in ditt DATA-TIPS från din bandstation eller diskdriver. Lista de rader som redovisas i bifogad listning och ändra dem. Nytillkomna rader skrivs med de radnummer som visas. Provkör programmet.

Då du kör igång det nu ändrade programmet ställs frågan: ''VAL: ENGELSKA (E) ELLER SVENSKA (S) LAG? '' Besvara frågan med E eller S. Väljer du E-svaret, fungerar ditt program precis som tidigare. Du har ju match-data fil på band eller disk. Väljer du svaret S, finns ingen färdig match-data fil — om du inte köpt ett färdigt program —. Det negativa svaret vid frågan om datafil finnes flyttar programmet till det avsnitt där filen skapas. Kör du bandstation uppmanas du skaffa ett band för ''engelsk matchdata och ett band för svensk matchdata''. Använder du floppyn, kan du behålla den engelska filen, eftersom filer med olika namn skapas. Du får på så sätt ett data-tidsprogram, två sekvensiella filer (MATCH-DATA och MATCH-DATA (S)) på samma diskett. Vid nyskapande av match-data fil, vilket kan ske i god tid före de svenska fotbollsserierna startar, gäller det att hålla sig till lagnamn, som datorn sedan känner igen. En komplett förteckning över de svenska lagen i allsvenskan, samt norra och södra division 2 serierna bifogas. Skapar du filen före seriestarten, matar du endast in lagnamnen samt besvarar alla övriga frågor med 0 (nollor). När inmatningen avslutats och filen stängts återstartas programmet. Du kan nu besvara matchdatafrågan med Y och t ex välja tabellutskrift och kontrollera dina lagnamn. Observera vidare, att för att ett lag skall kunna köras i tipssekvensen, måste laget i fråga ha spelat minst en match.

Fungerar allting korrekt, kan du nu spara undan programmet. Du har ett tipsprogram, som fungerar för både engelskt och svenskt tips. Programmet finns som tidigare att köpa från Ingvar Wihlborg. Tel: 0300—119 80. Se annons i denna tidning.

SVENSKA LAGNAMN FÖR DATA-TIPS:

ALLSVENSKAN:

AIK BRAGE ELFSBORG GEFLE GOTEBORG HALMSTAD HAMMARBY KALMAR FF MALMO FF NORRKOPING ORGRYTE OSTER

DIV 2 SODRA:

FROLUNDA GAIS GRIMSAS HACKEN HALMIA HELSINGBORG IFK MALMO KALMAR AIK LANDSKRONA LUND MJALLBY MYRESJO NORRBY TRELLEBORG

DIV 2 NORRA:

ATVIDABERG BROMMAP. DEGERFORS DJURGARDEN ESKILSTUNA IFK VASTERAS KARLSLUND NYKOPING OREBRO SANDVIKEN SKELLEFTEA SUNDSVALL VASALUND VASTERAS SK

(observera att Å Ä och Ö skrivs med A resp. O).

Ingvar Wihlborg
(371107-375)

LISTÄNDRINGAR OCH TILLÄGG FöR ATT KONVERTERA 'DATA-TIPS' TILL BÅDE ENGELSKA OCH SVENSKA MATCHER:

```
10 INPUT"VAL: ENGELSKA(E) ELLER SVENSKA(S) LAG";NT$
15 IFNT$="E"THENNT$="":AX=44:AT=21:A=3:GOTO30
20 IFNT$="S"THENNT$="(S)":AX=40:A=2:GOTO30
25 RUN
60 FORN=1TOAX
106 FORN=1TOAX:IFX$=L$(N)THENH=N:GOTO108
108 FORN=1TOAX:IFY$=L$(N)THENB=N:GOTO110
133 FORN=1TOAX
158 IFN=AXTHEN152
183 P(N)=V(N)*A+O(N):PRINTP(N)
225 FORN=1TOAX
235 FORN=1TOAX
240 IFG$(I)=L$(N)THENB=N:GOTO242
242 HZ=P(H):IFHZ<1THENHZ=1
243 BZ=P(B):IFBZ<1THENBZ=1
250 IFP(H)>P(B)THENET=ET+HZ/BZ
252 IFP(B)>P(H)THENTV=TV+BZ/HZ
1109 FORN=1TOAX:PRINT"J";"MATCH-DATA #";N:PRINT
1115 PRINT"INSL.  MAL ";:INPUT#9,W:PRINT#1,W:PRINT:PRINT#1,V-W:PRINT#1,X*A+Y
1125 PRINT#1,A:CLOSE1:CLOSE9:RUN
2001 IFA=2THEN2011
2010 PRINT"#2: DIVISION 2":PRINT:GOTO2020
2011 PRINT"#1: ALLSVENSKAN":PRINT
2012 PRINT"#2: DIV.2/SODRA":PRINT
2013 PRINT"#3: DIV.2/NORRA":PRINT:GOTO2036
2036 INPUT"TABELLVAL";TA
2037 ONTAGOTO2038,2040,2042
2038 PRINT"J";TAB(5)"ALLSVENSKAN":PRINTTAB(5)"——————————":PRINT:TX=1:TY=12:AT=1
1:GOTO2045
2040 PRINT"J";TAB(5)"DIV.2/SODRA":PRINTTAB(5)"——————————":PRINT:TX=13:TY=26:AT=
13:GOTO2045
2042 PRINT"J";TAB(5)"DIV.2/NORRA":PRINTTAB(5)"——————————":PRINT:TX=27:TY=40:AT=
13
2047 FORN=1TOAX :BB(N)=Ø:NEXTN:CC=Ø
2105 FORN=0TOAT
2200 REM:*******************************************
2210 REM:FöR DISKETTER GÄLLER DÄRUTöVER FöLJANDE
2215 REM:RAD 56   OPEN1,8,2,"MATCH-DATA"+NT$+",S,R":PRINT"MATCH-DATA LADDAS NU"
2220 REM:RAD 132  OPEN1,8,2,"@0:MATCH-DATA"+NT$+",S,W":PRINT "JMATCHRES. OCH....
.ETC"
2225 REM:RAD 1105 OPEN1,8,2,"MATCH-DATA"+NT$+",S,W":OPEN9,0
2230 REM:*******************************************

READY.
```

```
2 REM *****************
3 REM *   MADE BY      *
4 REM * HÅKAN THORSELL*
5 REM * NITAREGATAN 9 *
6 REM *58266 LINKÖPING*
7 REM * PROGRAM  FÖR  *
8 REM * VIC 64        *
9 REM *****************
10 PRINT CHR$(147)
20 PRINT"BOXNING"
30 PRINT"SKRIV DITT NAMN"
35 PRINT"";
40 INPUT S$
45 IF S$=""THEN S$="RASOPEN ANDERSSON"
50 PRINT"DU KAN SLÅ SWING,UPPERCUT,RASOP"
60 PRINT"S=SWING:U=UPPERCUT:R=RASOP"
65 PRINT"TRYCK EN KNAPP FÖR ATT BÖRJA"
70 GET A$:IFA$=""THEN 70
80 PRINT CHR$(147)
90 PRINT ""
91 PRINT ""
92 PRINT ""
93 PRINT ""
94 PRINT ""
95 PRINT ""
96 PRINT ""
97 PRINT ""
98 PRINT ""
100 PRINT"TRYCK DITT SLAG"
105 IFL=7 THEN 500
110 GET H$:IFH$=""THEN 110
120 IF H$="S"THEN200
130 IF H$="U"THEN300
140 IF H$="R"THEN400
150 GOTO 110
200 Q=INT(RND(1)*10)
210 IF Q>5THEN PRINT"TRÄFF MED SWING":L=L+1:T=T+1
215 IF Q<=5THEN PRINT"MISS MED SWING":L=L+1
230 GOTO 105
300 Q=INT(RND(1)*10)
310 IF Q>5THEN PRINT"TRÄFF MED UPPERCUT":L=L+1:T=T+1
315 IF Q<=5THEN PRINT"MISS MED UPPERCUT":L=L+1
330 GOTO 105
400 Q=INT(RND(1)*10)
410 IF Q>5THEN PRINT"TRÄFF MED RASOP":L=L+1:T=T+1
415 IF Q<=5THEN PRINT"MISS MED RASOP":L=L+1
430 GOTO 105
500 IF T>4 THEN 600
510 PRINT "";:PRINTS$" FÖRLORADE MATCHEN MOT VÄRLDSMÄSTAREN"
520 GOTO 610
600 PRINT "":PRINTS$" ÄR NY VÄRLDSMÄSTARE , HAN FICK IN "T" TRÄFFAR"
605 IFT>6THEN PRINTS$" HADE NYBÖRJARTUR"
610 PRINT"VILL DU BOXAS EN GÅNG TILL"
620 GET P$:IFP$=""THEN 620
625 IF P$="N"THEN 650
630 IF P$="J"THEN RUN
640 GOTO 620
650 END

READY.
```

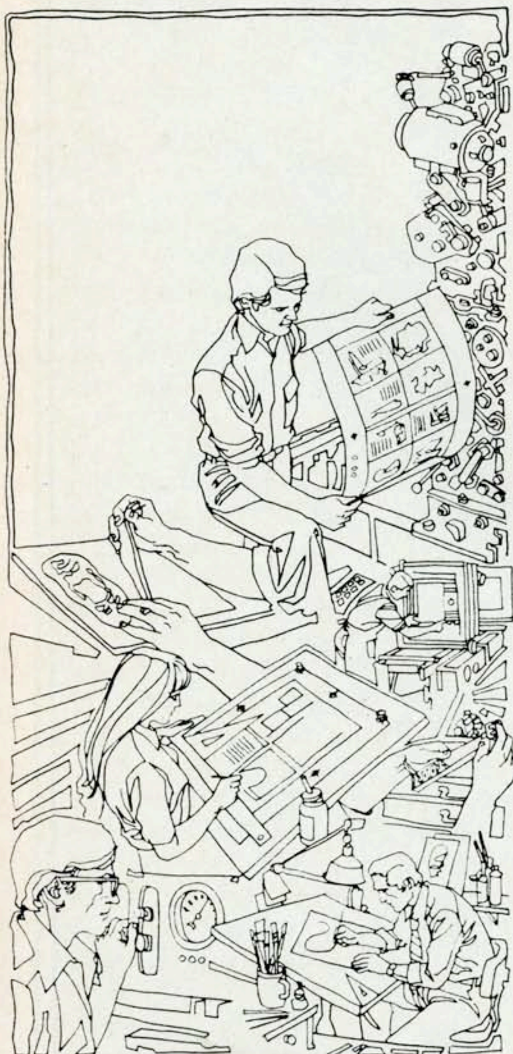
```
0 RESTORE:CLR
1 PRINT"□"
2 REM OMGJORT 1983 16/12 AV HAKAN O ELISABETH THORSELL
3 REM URSPRUNGLIGEN FRAN VIC RAPPORT
5 PRINT"**** VILKEN VECKODAG ****"
10 INPUT"VILKET AR";A
11 INPUT "VILKEN MANAD";B
12 INPUT "VILKET DATUM";C
30 FOR I =1 TO 2
40 D=A-1753
50 S=D*365.25
55 IF A<1753 THEN PRINT "ARET MASTE VARA EFTER 1753"
56 IF B>12 THEN PRINT" ETT AR HAR HöGST 12 MANADER"
57 IF C>31 THEN PRINT"EN MANAD HAR HöGST 31 DAGAR"
60 IF D>47 THEN S=S-1
70 IF D>147 THEN S=S-1
75 S=INT(S)
80 NEXT I
85 IF A<1752 THEN GOTO 89
86 IF B>13 THEN GOTO 89
87 IF C>32 THEN GOTO 89
88 GOTO 90
89 PRINT"DET HäR FAR DU GöRA OM!":GOTO 10
90 FOR J=1 TO 12
100 READ D
110 IF J=B THEN 132
120 S=S+D
130 NEXT J
132 IF A=1800 OR A=1900 THEN 145
135 IF J=2 AND A/4=INT(A/4)THEN D=29
140 IF J>2 AND A/4=INT(A/4) THEN S=S+1
145 GOTO 150
150 DATA 31,28,31,30,31,30,31,31,30,31,30,31
152 PRINT:PRINT
155 PRINT"DEN";C;"/";B;"AR";A;"äR EN ";
160 S=S+C
163 IF C>D THEN GOTO 1
165 IF C>D THEN PRINT "LäR DIG MANADERNA!"
170 S1=S-INT(S/7-0.1)*7
180 ON S1 GOSUB 201,202,203,204,205,206,207
190 GOTO 210
201 PRINT"MANDAG":RETURN
202 PRINT"TISDAG":RETURN
203 PRINT"ONSDAG":RETURN
204 PRINT"TORSDAG":RETURN
205 PRINT"FREDAG":RETURN
206 PRINT"LöRDAG":RETURN
207 PRINT"SöNDAG":RETURN
210 PRINT"■VILL DU FORTSäTTA (J/N)?"
211 GETL$:IFL$=""THEN 211
212 IF L$="J"THEN RUN
213 IF L$="N"THEN 220
214 GOTO 211
220 PRINT"□SLUT!"
230 END

READY.
```

Rapport från Skäftingbackens Datorörnar

Skäftingebackens Unga örnar hade ett problem gemensamt med alla barn och ungdomsorganisationer. När medlemmarna växte upp så tröttnade de på verksamheten, och slutade.

För att behålla medlemmarna så beslöt styrelsen att starta en datorgrupp. Jag fick uppdraget att sköta det praktiska. Först gällde det att välja dator. Vid en första granskning hur mycket dator man får per krona så föll genast ABC och Apple bort. Sedan gällde det att försöka gissa vilka datorer som kan finnas kvar på marknaden om 5 år. Det är inte så roligt att köpa en dator, om det inte finns tillbehör eller reservdelar till den efter några år. Då återstod det bara VIC och Sinclair. Att välja mellan en dator och en leksak var inte svårt. Det blev VIC. Styrelsen beslöt att köpa in en VIC 20. Själv köpte jag en 64. Datorgruppen fick namnet Skäftingebackens datorörnar. För att få in pengar till löpande kostnader så tog vi ut en gruppavgift på 100:— per person och termin. Många olyckskorpar kraxade och trodde att ingen skulle betala det. Men alla 10 platserna blev genast fulltecknade. Vi lyckades ragga upp två skraltiga svartvita TV som vi bar ner i Unga örnars lokal. När det gällde studiemateriel så var det värre. Det finns hur mycket dåligt materiel som helst för vuxna. Men inget som är lämpligt från 10 år och uppåt. Så jag fick göra eget materiel. Vi började så smått med att ha möte en kväll i veckan. 1 timme utbildning i programmering, och en timme spel. Snart så kom medlemmarna och lånade datorn själva. För att minska slitaget på kontakterna så monterade vi upp anläggningen på en spånskiva. Ett vanligt handtag från en kökslåda gjorde det lättare att bära. Nu var datorn utlånad strängt taget varje kväll i veckan.



Till en början var det mest spel för hela slanten. Men snart började några att programmera på egen hand med hjälp av manualen och de handböcker vi hade köpt. De aktiva fick egna kassetter som de kunde spara programmen på. Nu började verksamheten svälla ut för mycket. Det var för många intresserade, och bara två datorer.

Vi ansökte om bidrag till utrustning av lokalen. Det blev genast backslag från fritidsförvaltningen. Utrustningsbidrag skall användas till inköp av möbler, gardiner, färg och annat till lokalen. Inte till annat. Något nytt bidrag skulle vi inte få förrän om 10 år.

Efter mycket förhandlande så fick vi köpa datorutrustning i stället. Nu hade vi 16 000 att sätta sprätt på. Vår leverantör VIC-center ställde som alltid upp för oss. Vi lämnade in en beställning. Betalningen skulle ske genom faktura. Då slapp vi problemet med att springa omkring med en massa pengar i fickan. Det blev genast en 64:a, bandstation, diskettenhet, skrivare och en massa utrustning till det. För att förhindra stöld, så har vi monterat stöldlarm, galler för fönstren, förstärkt dörrarna, och en massa annat. Skall någon obehörig ta sig in så får de använda dynamit. Kofot räcker inte. Nu är det full verksamhet i lokalen. Som mest är vi 15 medlemmar. Till detta kommer det tillfälliga besökare som får pröva på att ratta datorerna.

Många vuxna undrar vad det är för nytta med att kunna programmera ett spel. Det är ingen som har fått anställning bara för att de kan programmera i BASIC.

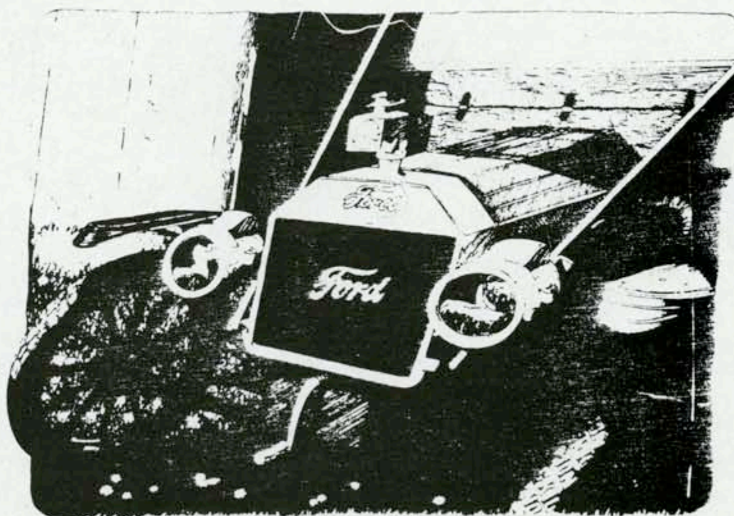
Jag brukar be dem titta på samhället i dag. Det är många som är direkt beroende av bilen, såsom tillverkare, reparatörer, försäljare, drivmedelsförsäljare och annat. Det kommer att bli samma med datorn.

När det gäller datorer har vi ännu inte kommit in i T-fordstadiet. Fundera hur samhället kommer att se ut om 10 år. ALLA kommer att vara beroende av datorer. När det gäller BASIC så är det ett nybörjarspråk. Först så måste man begripa hur datorn och ett program fungerar. Därefter kan man gå över till strukturerad programmering och filhantering. Alla våra stora författare började med att skriva; FAR ÄR RAR MOR ÄR EN ORM. De började inte i sjunde klass med att författa romaner.

En positiv sida med datorer är att intresset för skolan ökar. Nu inser de att de måste lära sig engelska. Annars kan de inte läsa alla krångliga bruksanvisningar som medföljer apparaterna. Utan mattekunskaper så blir det inga bra program. Skrivmaskinsträning kommer på köpet.

Så vi i Skäftingebackens datorörnar har bara positiva erfarenheter av datorer. Det bästa råd vi kan ge är: Starta en grupp själva.

Skäftingbackens datorörnar
Alf Olsson



Starta datorgrupper

Alldeles förmånga sitter hemma och knappar i sin ensamhet. Mest beroende på att de inte har någon att dela sitt intresse med. Varför inte starta en förening och vara tillsammans med andra hackers. Problemen kan verka många och oöverstigliga. Men de går att övervinna.

1. Hur få medlemmar. Sätt upp en lapp på skolornas anslagstavlor. När ni är ca 5 st så kan ni bilda en interimsstyrelse. Stadgar för föreningen kan ni planka i VIC-tidningen nr 4.

2. Se till att få medlemmar under 25 år. Kommunens fritidsförvaltning lämnar aktivitetebidrag om det är 5 medlemmar i åldern mellan 7 och 25 år som håller en planerad verksamhet om minst en timme. Hur stort bidraget är varierar mellan varje kommun. Lägst är det statliga bidraget på 10:— per sammankomst. Upp till 50:— i Stockholms kommun. En förutsättning är att en majoritet av medlemmarna i hela föreningen är under 26 år. Tanken är att det skall vara en ungdomsorganisation.

Tag kontakt med kommunens fritidsförvaltning, eller med sociala servicecentralen så kan de lämna mer upplysningar.

Kommunen kommer troligen att kräva att ordföranden eller kassören är med på ett informationsmöte som kommunen anordnar i bidragsbestämmelser.

När ni har haft cirka 20 sammankomster. Då kan ni ansöka om startbidrag för föreningen.

Nu gäller det att ordna en lokal. Många föreningar har en lokal som de utnyttjar dåligt. För att få behålla sina hyresbidrag så måste de ha minst 4 sammankomster i veckan. Många ligger på håret. Prata med styrelsen om ni inte kan dela på lokalen. Har de 2 sammankomster, och ni tre per vecka. Då klarar bägge gränsen för att ha egen lokal.

Nu bör lokalfrågan vara löst. Om kommunen bidrar med 90% av hyran. Då bör det vara lätt att förhandla om hur stor del varje förening skall betala själv. Nu behöver ni utrustning till lokalen. Många kommuner lämnar bidrag till utrustning av lokalen. Genom att ni har startat en datorförening så är det naturligt att ni möblerar med skrivare, diskett och datorer.

Är ni ett gäng över 14 år så kan ni starta en studiecirkel i datorer. Lägsta gräns är 5 deltagare under 14 år. Kontakta ABF så kan de berätta allt. Cirkelledaren får betalt för att ställa upp.

När det gäller TV-apparater så kan ni gå runt i grovsoprum och containrar och leta. Sätt upp lappar på portarna att ni vill ha begagnade fungerande TV-apparater. Är ni en förening så är folk piggare på att ställa upp. Prata med TV-handlarna om ni inte kan få en inbytesapparat som de ej kan sälja.

När ni väl har startat en fungerande förening, då är många dörrar öppna. Ni kan få låna skolornas datorer. Eller så kan ni få tillträde till många databaser. Då är det inte en enskild person, utan en förening som står för ev skadegörelse och alla kostnader.

Lycka till med startandet av datorgrupper.

Skäftingebackens datorörnar

```
READY.

0 REM ***************
1 REM * MASTER MIND *          * AV JOHAN     *          * NASLUND       *          * 1983-1
2-27  *
2 REM ***************
3 REM"■■■■■■■
5 GOSUB5000
6 GOTO6020
20 M$="*■■ ● ● ● ●  ■*    *":B$="MASTER MIND":POKE36879,78:POKE36878,15
30 Y%=3:X%=5:P%=7680:F=38400:X1%=15:S1=36874:S3=36876:SL%=0
40 FORT=0TO3:K%(T)=0:K1%(T)=0:K2%(T)=0:NEXTT
100 REM*** SPELPLAN **
110 PRINT"■■■*M*■■■ ● ● ● ● ■"
120 FORT=2TO11:PRINT"■* *":PRINT"*"MID$(B$,T,1);M$:NEXTT
200 REM**** KODEN ****
210 FORT=0TO3:K1%(T)=INT(RND(1)*8)+1:NEXTT
220 GOTO420
300 REM** GISSNING ***         ** H. PROG. ***
310 GOSUB8010
320 PE%=PEEK(P%+X%+22*Y%):IFPE%<128THENPOKEP%+X%+22*Y%,PE%:POKEF+X%+22*Y%,6:GOTO
350
330 POKEP%+X%+22*Y%,PE%-128
350 IFA$="←"THENGOSUB1000:GOTO420
355 IFA$="■"THEN410
360 A%=VAL(A$)
370 IFA%<1ORA%>8THEN310
380 POKEP%+X%+22*Y%,48+A%
390 T%=(X%-5)/2
400 K%(T%)=A%
410 X%=X%+2:IFX%>11THENX%=5
420 IFSL%=1THEN2100
425 IFSL%=2THEN2000
430 POKES3,180+A%*3:POKEP%+X%+22*Y%,PEEK(P%+X%+22*Y%)+128:POKES3,0
500 GOTO300
1000 REM*** JAMFOR ***          *** KODER  ***
1020 FORT=0TO3:K2%(T)=K1%(T):NEXTT
1040 FORT=0TO3
1050 FORR=0TO3
1060 IFK%(T)=K2%(R)THENIFR=TTHENGOSUB3000
1070 NEXTR,T
1080 IFX1%>18THENSL%=2:RETURN
1100 FORT=0TO3:FORR=0TO3
1110 IFK%(T)=K2%(R)THENIFR<>TTHENGOSUB3040
1120 NEXTR,T
1150 X1%=15:X%=5:Y%=Y%+2:FORT=0TO3:K2%(T)=0:K%(T)=0:GETA$:NEXTT
1180 IFY%>22THENSL%=1:RETURN
1200 RETURN
2000 REM***  SLUT  ***
2002 POKE36879,27
2005 GOSUB4000
2008 PRINT"■■■BRA "S$(SP%)
2010 PRINT"■■  DU KLARADE DET PA     "((Y%-3)/2)+1" FORSOK"
2015 FO%(SP%)=((Y%-3)/2)+1
2020 PRINT"■■■KODEN VAR:"
2030 PRINT"■■■";:FORT=0TO3:PRINTK1%(T);:NEXTT
2035 PRINT
2038 IFSP%<AT%THEN6000
2039 PRINT"■■■":GOSUB8000:GOSUB7000
2040 PRINT"■■■■  ETT SPEL TILL ? ■"
2050 GETA$:IFA$="J"THENRUN
2060 IFA$="N"THENEND
2070 GOTO2050
2100 REM**** SLUT ****          **** MISS ****
2102 POKE36879,110:GOSUB4100:GOSUB8020:POKE36879,26
```

```
2105 PRINT"■■■■SYND "S$(SP%)
2110 PRINT"■■■■DU KLARADE DET INTE"
2120 GOTO2020
3000 REM** RATT FARG**          ** PA RATT  **          ** PLATS     **
3010 POKEP%+X1%+22*Y%,90:POKEF+X1%+22*Y%,0
3020 K2%(R)=-2:K%(T)=-1:X1%=X1%+1:RETURN
3030 REM** RATT FARG**          ** PA FEL   **          ** PLATS     **
3040 POKEP%+X1%+22*Y%,90:POKEF+X1%+22*Y%,1
3050 K2%(R)=-2:K%(T)=-1:X1%=X1%+1:RETURN
4000 REM** GLADLJUD **
4010 FORT=1TO5:FORR=150TO220:POKES3,R:NEXTR,T
4015 POKES3,0
4020 RETURN
4100 REM** SUR-LJUD **
4110 FORT=1TO80:POKES1,207-T:FORR=1TO10:NEXTR,T
4120 RETURN
5000 REM*** INFORM ***
5002 POKE36879,76
5005 PRINT"■■******■MASTER MIND■******"
5020 PRINT"■■JAG TANKER PA EN KOD  PA FYRA SIFFROR (1-8) SOM DU SKA KNACKA"
5025 PRINT"
5030 PRINT"■EFTER VARJE GISSNING  VISAR JAG HUR MANGA   SOM SITTER RATT:■■■"
5040 PRINT"■OCH HUR MANGA SOM     SITTER PA FEL PLATS:■■■"
5045 PRINT"■■AVSLUTA RADERNA MED'←'
5048 PRINT"■ANVAND CRSR ← FOR ATT FLYTTA UTAN UTSKRIFT■"
5050 PRINT"■DU HAR 10 FORSOK "
5060 PRINT"■■■■";:GOSUB8000
5100 REM***  ANTAL  **          *** SPELARE **
5105 POKE36879,15:PRINT"■■■■■HUR MANGA SPELARE(1-4)"
5110 INPUT"■";AT%
5115 IFAT%<1ORAT%>4THEN5100
5120 FORT=1TOAT%
5130 PRINT"■■■VAD HETER SPELARE"T
5135 PRINT"■HOGST 10 BOKSTAVER"
5140 INPUT"■■";S$(T)
5145 IFLEN(S$(T))>10THEN5130
5150 NEXTT:RETURN
6000 REM** VEMS TUR **
6010 PRINT"■■■":GOSUB8000
6020 SP%=SP%+1:IFSP%=1THENPRINT"■■■■OKEJ "S$(SP%):PRINT"DU BORJAR !":GOTO6030
6025 PRINT"■■■■OKEJ "S$(SP%):PRINT"NU AR DET DIN TUR"
6030 PRINT"■■■TRYCK IN EN KNAPP NAR DU AR BEREDD !"
6050 GOSUB8010
6060 GOTO20
7000 REM** RESULTAT **
7001 IFAT%<2THENRETURN
7015 PRINT"■■"TAB(6);
7016 FORT=1TO8:PRINT"■"MID$("RESULTAT",T,1);:FORR=1TO150:NEXTR,T
7017 PRINT
7018 IFAT%<3THENPRINT"■■NU HAR BADA SPELAT EN GANG VAR■":GOTO7030
7020 PRINT"■■NU HAR ALLA SPELAT EN GANG VAR■"
7030 FORT=1TOAT%
7040 IFFO%(T)=0THENPRINT"■"S$(T)" LYCKADES INTE":GOTO7060
7050 PRINT"■"S$(T)" GJORDE"FO%(T)"FORSOK"
7060 NEXTT
7070 RETURN
8000 PRINT"TRYCK IN EN KNAPP !"
8010 GETA$:IFA$<>""THENRETURN
8020 GOTO8010

READY.
```

Text och grafik samtidigt

Tricket går ut på att man gör om ett text-tecken till ett grafik-tecken. Så här går det till: när man skriver något på bildskärmen, står detta tecken i bildskärmens RAM. Detta tecken kommer man endast åt med PEEK och man får då bildskärmskoden. Exempel: uppe till vänster står tecknet ''A''. Trycker man nu in ''PRINT PEEK (1024)'', så får man som resultat talet 1. Med detta tal kan datorn hitta tecknet i teckengeneratorn.

Ett tecken består av en 8×8-matris (se bild). En rad i denna matris kan man framställa med ett tal. Är t ex bit 7 satt, så får man talet 128 (2ù7). Är flera bitar satta, så får man motsvarande tal genom addition. Bilden visar detta med bokstaven A. Tecknet står kodat på detta sätt i teckengeneratorn. Nu behöver man bara bildskärmskoden för att läsa ut motsvarande tecken ur teckengeneratorn. Då man inte kommer åt teckengeneratorn från BASIC, flyttar maskinkodsprogrammet i rad 1 till 6 teckengeneratorn till slutet av BASIC-RAM:et (adress 34816 och framåt). Samtidigt sätts slutet på BASIC-RAM:et till denna adress för att skydda mot överskrivning. Till programmet: huvudprogrammet, som flyttar tecknet från lagringsområdet 1024—2023 till området 8192—16192 utförs som underprogram (rad 1000 till 1050). Det anropas i rad 50, som överför följande parametrar: i ZE står raden (0 till 25, 0 betyder överst) och i SP står kolumnen (0 till 39, 0 betyder vänstrast), varifrån texten ska stå. Texten står i variabeln TES. Med hjälp av STRS kan även numeriska variabler skrivas ut, t ex. TES = STRS(A). I VF står tecknets färg och i HF bakgrundsfärgen. För färgerna används talen 0 till 15 som vid utnyttjande av färg-RAM:et (o = svart, 1 = vitt o s v).

När underprogrammet anropas, skrivs texten ut i rad 1000. På bildskärmen står först färgrika fyrkanter, då bildskärms-RAM:et ju används som färg-RAM. Rad 1010 beräknar lagringsläget i färg-RAM:et (1024 till 2023) och den därtill hörande adressen i bildskärmens RAM (8192 till 16192). I rad 1030 läses tecknen av och början på kodningen i den förskjutna teckengeneratorn beräknas. Rad 1040 slutligen, skriver ut detta tecken på skärmen. För övrigt tilldelas variabeln CPS styrtecknen 1äCURSOR HOME + 24*CURSOR DOWN. Därmed når man lätt varje position på skärmen (se rad 1000). Det fungerar med grafiska tecken också. Tänkvärt är att man ändrade i den förskjutna teckensatsen och på så sätt fick t ex grekiska tecken. Till sist: detta program fungerar bara i högupplösnings-moden, då datorn fortfarande försöker skriva ut orginaltecknen.

Så fungerar en Joystick

Joysticken verkar bli ett ''kultobjekt'' för en hel generation. Fast man fascineras över den, är konstruktionen enkel. Ett par passiva elektronikkomponenter och lite mekanik är hela hemligheter.

''Joystick'' kallar man en apparat med en styrspak, som användaren kan flytta i minst fyra riktningar och som överför styrsignaler som motsvarar spakens läge till den anslutna datorn. Det finns naturligtvis mer än bara en sorts joystick. Inte bara kvalitén, utan även konstruktionsprincipen skiljer dem åt.

Styrspaken är dock typisk. Ena änden är förbunden med en låda, på ett sätt som tillåter ca: 30 graders avvikelse i varje riktning (det skiljer sig från joystick till joystick). Själva styrspaken kan vara konisk, rak eller böjd, platt eller greppvänligt formad och gjord av trä, plast eller metall. Hur joysticken ser ut har dock väldigt lite att göra med funktionsprincipen. Men för att joysticken ska fungera bra även i snabba spel, är dessa skillnader avgörande.

Digitala Joysticks känner bara riktningen

Vad som intresserar oss är de olika funktionsprinciperna. Man har två huvudtyper: de ''binära'' som bara känner t ex ''höger'' eller ''inte höger'' och de ''proportionella'' som känner hur långt från mittläget styrspaken befinner sig. För enkelhets skull kallar vi dem digitala respektive analoga joysticks.

De digitala har binära utgångssignaler och består i det inre av ett litet antal strömbrytare (minst 4), som är ordnade som en cirkel eller ett kors. Dessutom finns en rund eller fyrkantig basplatta fastsatt på styrspakens nedre del. När man rör på styrspaken kan denna platta med hjälp av piggar trycka ner en eller högst två strömbrytare. Då aldrig mer än två strömbrytare kan tryckas ner samtidigt, får anordningen exakt dubbelt så många urskiljbara riktningsangivelser som det finns strömbrytare. Eftersom färre än fyra strömbrytare är vettlöst, har en joystick minst åtta riktningar. Det finns joysticks med åtta strömbrytare och alltså sexton riktningar. De kan dock inte anslutas till alla datorer, då det krävs mer hårdvara för att utvärdera signalerna. Användningsvärdet ökar naturligtvis ju fler riktningar joysticken har. Oftast finns även skjutknapp, som dock inte har med styrningen att göra.

Strömbrytaren består tyvärr ofta bara av en välvd metallplatta över två elektroder. När metallplattan trycks ner, förbinder den de två elektroderna och sluter kontakten. Denna konstruktion har två stora fördelar för tillverkaren: den är billig att tillverka och sörjer dessutom för styrspakens fjädring, när man släpper styrspaken fjädrar ju metallplattan tillbaks och för styrspaken till sitt ursprungsläge. Nackdelarna för användaren är dock ganska graverande: plattan kan gå sönder av utmattning efter en tid och förlora fjädringskraft (styrspaken återtar ej ursprungsläget) eller glida (kontakt blir svårare). Är inte kontakterna förbyllda uppstår ännu en fara. Då kan nämligen metallen oxidera och bara vid väldigt hårt tryck kan rätt signal erhållas.

De erhållna signalerna matas in parallellt i datorn, som tolkar signalerna som binära tal. Varje läge på styrspaken motsvarar exakt ett binärvärde. Detta lagras i en reserverad minnescell (dess adress finner man ofta i handböckerna). Så här långt behandlar nästan alla datorer joysticksignalen på samma sätt. Hur ett program sedan behandlar detta värde skiljer sig lika mycket som programmen själva. För det mesta har programmet en tabell med olika värden som den jämför med den reserverade minnescellen. Ett andra värde i tabellen tala om hoppadress för fortsatt programexekvering. Teoretiskt kan en rörelse uppåt med styrspaken utan vidare motsvara cursorförflyttning åt höger. Vad som ska hända vid en viss spakrörelse bestäms alltså helt av programmet.

Av den digitala joystickens konstruktionsprincip framgår en, för användaren viktig egenskap: då varje läge på styrspaken motsvarar ett binärt värde, ändrar inte styrkan man trycker med detta värde. Då vet inte datorn hur hårt användaren tryker. I t ex ett rymdspel är det alltså meningslöst att försöka få ett rymdskepp att röra sig fortare genom att trycka hårdare på styrspaken.

Dyr men fin: Analoga Joysticks

Saken ser annorlunda ut hos den analoga joysticken. Denna oftast mycket dyra ''lyxspak'' består istället för strömbrytare av två potentiometrar (mekaniskt ändringsbara elektriska motstånd), vars rotationsaxlar står i rät vinkel i förhållande till varandra — motsvarande koordinataxlar på en tvådimensionell bildskärm. I bild 2 kan man se en analog joystick med två vanliga vridpotentiometrar. Styrspakens upphängning är något problematisk, då den dels ska vara rörlig i alla riktningar, dels ska överföra vridrörelser till de båda potentiometrarna. För det mesta använder tillverkaren halvcirkelformade böjda ''klämmare'' som i sin tur är rörliga runt en axel, nämligen en av potentiometeraxlarna.

Det finns för övrigt ansatser till elegantare lösningar på det mekaniska problemet med lysdioder och fotoceller som ändringsbara motstånd. Vid spakrörelse skulle dessa förmörkas beröringsfritt. I princip kan varje fysisk effekt som orsakar motståndsändring omvandlas på ett eller annat sätt och möjligheterna är långt ifrån uttömda. Här finns en chans även för amatörbyggare och hobbyuppfinnare.

Uppbyggandet av en analog joystick är dyrt framförallt på grund av den påkostade mekaniken. För användaren är det viktigt att veta om han överhuvudtaget kan ansluta en analog joystick till sin dator, det krävs nämligen en analog ingång för det. Denna måste ha en kontakt som levererar en spänning till joysticken. Denna spänning reduceras proportionellt mot styrspakens avvikelse och skickas tillbaka till datorn. Då krävs naturligtvis två kanaler in och två kanaler ut ur joysticken.

Analoga värden måste omvandlas till digitala med A/D-omvandlaren

I motsats till den digitala joystickens signaler, kan inte datorn använda den analoga signalen utan föregående signalbehandling. Därför måste datorn ha en A/D-omvandlare inbyggd i ingången. Denna A/D-omvandlare omsätter ett analogt spänningsvärde till ett digitalt. Det digitala värdet kan vidarebefordras från omvandlaren som parallella eller seriella binära signaler. Dessa signaler lagras i en reserverad minnescell, precis som för digitala joysticks. Vad som sedan händer avgörs av det aktuella programmet.

Hur bra upplösningen på riktningarna och avvikelserna blir, beror främst på hur många bitar A/D-omvandlarens digitala signal består av. Levererar en omvandlare t ex en 8-bit-signal, så kan den därmed framställa 256 olika värden. Med två kanaler motsvarar det 512 värden för en tvådimensionell rörelse. Jämför man det med de 8 värdena för en normal digital joystick, förstår man fördelen med den analoga konstruktionen. Med den analoga styrspaken kan man t ex även låta styrkan som användaren trycker med användas för hastighetsreglering av ett flygande tefat i ett ''star war''-spel. Den som en gång njutit av att spela TV-spel på detta sätt, får allt svårare att nöja sig med den enkla digitala joysticken.

Lägg in Joystick i dina spel

Många som konstruerar spel låter spelet styras av tangenter i stället för med joystick. En nackdel med detta är att datorn lätt blir misshandlad i stridens hetta. Även om du tar det försiktigt med datorn så finns det många som tycks tro att skotten går längre, ju hårdare de trycker på tangenterna.

Om du har möjlighet så lägg in joyfunktioner i dina spel. Går det att lista spelet, så kan du snart klura ut hur du skall göra.

Följande instruktioner avser VIC-20.

Pröva följande grundprogram.

```
1/  POKE 37154,127
1// A = PEEK (37137)
11/ B = PEEK (37152)
2// IF NOT A AND 4 THEN PRINT "UPP"
21/ IF NOT A AND 8 THEN PRINT "NER"
22/ IF NOT A AND 16 THEN PRINT "VÄNSTER"
23/ IF NOT B AND 128 THEN PRINT "HÖGER"
24/ IF NOT A AND 32 THEN PRINT "FIRE"
3// GOTO 1//
```

Rad 10 ställer om ingångarna så att joystick HÖGER kan användas. Den omställningen behöver bara anges en gång i programmet. Förslagsvis lägger du den i början av programmet.

När man vill ha en ny spelomgång så är det bekvämare att trycka på FIRE i stället för på en tangent. Det gör du så här.

```
4/// PRINT "EN GÅNG TILL"
4/1/ PRINT
4/2/ PRINT "( J/N )
4/3/ GET XS
4/4/ IF XS = "J" THEN (lämpligt startnummer)
4/5/ IF XS = "N" THEN 5///
4/6/ A = PEEK (37137)
4/7/ IF NOT A AND 32 THEN (lämpligt startnummer)
4/8/ GOTO 4/3/
5/// PRINT "HEJ DÅ"
5/1/ NEW
```

Rad 4/3/ söker av tangentbordet. Är J nertryckt så går datorn till det nummer du har angett. Är N nertryckt så avslutas spelet, och minnet raderas i rad 5/1/.

Är ingen tangent nedtryckt så kontrollerar rad 4/7/ om FIRE är nedtryckt på joyen. Rad 4/8/ gör att slingan börjar om från början på rad 4/3/.

Om du har lagt in en instruktionstext i början av programmet, så uppstår det problemet att du inte hinner släppa FIRE knappen förrän texten har blinkat förbi. Det löser du genom att lägga in en fördröjningsslinga i programmet.

```
1// PRINT "instruktionstext"
11/ PRINT
12/ PRINT "KLAR ATT SPELA ?"
13/ FOR A = 1 TO 9///: NEXT
14/ A = PEEK (37137)
15/ IF NOT A AND 32 THEN radnummer för spelstart
16/ GET AS
17/ IF AS    "" THEN radnummer för spelstart
18/ GOTO 14/
```

Rad 13/ är en fördröjning så att det inte blir "dubbelslag" med FIRE knappen. 15/ känner av om FIRE är nedtryckt. 17/ kontrollerar om någon tangent är nedtryckt, (om AS är större än ingenting. I så fall fortsätter datorn på den rad du har angett. Annars så fortsätter slingan på rad 14/.

Får du problemet att datorn är så snabb att det blir dubbelslag. Då kan du lägga in en fördröjningsslinga innan datorn söker av tangenter och joystick.

```
3/// utfört instruktionen
3/1/ GOTO 4///
4/// FOR A = 1 TO 1//: NEXT
4/1/ avläsningsslingan
41// GOTO 4/1/
```

Lycka till med ditt progamerande.

Skäftingebackens datorörnar
Alf Olsson

VIC-rapport är bra!

VIC-rapport är i stort en bra tidning.

Det framgår av den läsekretsundersökning som genomfördes i VIC-rapport nr 5/6.

Här nedan kan Du själv läsa vad VIC-rapports läsare tyckte. Det är medelvärden som redovisas.

Elva omdömen om innehåll resp. text sattes på en femgradig skala.

	Innehållet	Text
LEDARE	3.4	3.9
NYBÖRJARSKOLAN	4.0	4.1
KURS I STRUKT. BASIC	3.2	3.3
ASSAMBLERSKOLAN	3.7	3.3
COMANKURSEN	3.3	3.3.
PROGRAMMERINGSTEKNIK	4.1	3.8
NYHETSPRESENTATIONER	3.8	4.1
FRÅGOR OCH SVAR	3.5	3.9
KAPTEN STORÖRA	3.0	3.8
ANNONSER	3.7	4.2
SMÅTIPS	3.9	4.0
GENOMSNITTLIGT MEDELVÄRDE	3.6	3.8

Av ovanstående sammanställning kan vi utläsa följande: Innehållsmäsigt får PROGRAMMERINGSTEKNIK och NYBÖRJARSKOLAN goda omdömen (4.1 resp. 4.0).

Likaså ser vi att SMÅTIPS och NYHETSPRESENTATIONER har högt läsvärde (3.9 resp. 3.8). ANNONSER och ASSEMBLERSKOLAN noterar båda 3.7 för innehållet.

På den femgradiga skalan blir helhetsomdömet för innehållet 3.6. Läsbarheten frågade vi också om, och där ser redaktionen att tidningen genomgående har en god läsbarhet (3.8).

Högsta läsbarheten har ANNONSER noterade 4.2 i läsbarhet på en femgradig skala måste annonsmakarna uppfatta som ett gott betyg på sitt hantverk. Grattis.

Bakläxa får tidningen på de avfotograferade listorna, vilka främst förekommer i kursavsnitten.

Redaktionen hoppas kunna förbättra reproduktionen av dessa listor. Redaktionen drar den slutsatsen att läsarna tycker om VIC-rapport, men vill ha en förbättring avseende listor.

Många läsare har också befogat brev med förslag, vilka redaktionen kommer beakta efter ytterligare studium.

Redaktionen tackar alla som har sagt sin mening och hoppas att vi tillsammans med er kan göra en ännu bättre VIC-rapport i framtiden.

Thomas Flink
Redaktionen



Experiment med VIC-20

En VIC-20 kan förvisso användas till mer än spel. Kanske datorns stora användningsområde kan vara experiment och försök. Det relativt lilla minnet är här knappast något hinder eftersom många s k vetenskapliga program är föredömligt korta. Att lösa svåra problem på ca 500 bytes är inte ovanligt.

Jag skall här redovisa några experiment med slantsingling m a o statistiska experiment. De experimentserier jag redovisat här har jag inte funnit redovisade förut. Utan en dator är de nämligen i praktiken omöjliga att genomföra. Jag har alltså befunnit mig i en situation som statistikforskare tidigare bara kunnat drömma om och knappast det.

Varför frågor nu någon. Varför gör man nu detta, som jag skall redovisa. Ja själv anser jag det vara en blandning av konst och högst grundläggande vetenskap. Mina experimentförutsättningar har varit helt ideala, varför detta även är en redovisning av det *perfekta* experimentet. Tack vare VIC-20 har jag kunnat genomföra en försöksserie som helt manuellt aldrig skulle ha genomförts.

Experimentserie 1

Min första experimentserie gällde fördelningen av krona och klave vid olika antal kast. Varje seriestorlek (10, 100, 1000 och 10000 kast) testades 30 gånger. Sammanlagt simulerades således 333 3000 slantsinglingar. Resultatet redovisas i tabell 1.

Tabell 1. Fördelningen av krona och klave vid 30 experiment för varje seriestorlek

Seriestorlek	Antal krona	Antal klave
10	5,4	4,6
100	49,5	51,5
1000	491,8	508,2
10000	4974,8	5025,2

Program för experiment 1

```
10 PRINT "  "
20 H = 0
30 L = 0
40 INPUT "ANTAL KAST"; A
50 PRINT "TA DET LUGNT!"
60 PRINT "DETTA TAR KANSKE"
70 PRINT "LITE TID, SER DU"
80 FOR J = 1 TO A
90 K = INT (RND (1) * 2) + 1
100 IF K = 1 THEN H = H + 1
110 IF K = 2 THEN L = L + 1
120 NEXT J
130 PRINT "  "
140 PRINT H; "KRONOR"
150 PRINT L; "KLAVE"
160 END
```

Experimentserie 2

Nästa experimentserie gällde hur många kronor i rad som maximalt kom upp vid olika seriestorlekar. Varje seriestorlek testades 10 gånger. Resultatet visas i tabell 2.

Tabell 2. Antal krona i rad som mest vid 10 experiment för varje seriestorlek

Seriestorlek	Maxantal i rad
10	5
100	7
1000	12
10000	14

Program för experiment 2

```
10 PRINT "  "
20 A = 0
30 P = 0
40 INPUT "ANTAL KAST"; B
50 PRINT "TA DET LUGNT"
60 FOR L = 1 TO B
70 K = INT (RND (1) * 2) + 1
80 IF K = 1 THEN A = A + 1
90 IF K = 1 AND A  P THEN P = A
100 IF K = 2 THEN A = 0
110 NEXT L
120 PRINT P; "KRONOR I RAD"
130 PRINT "SOM MEST"
140 END
```

Slutord

Jag har här visat på ett kaske annorlunda sätt att bedriva stimulerande verksamhet med hjälp av en VIC-20. Som synes är det utnyttjade programmen mycket korta och knappast speciellt komplicerade trots att de löst högst besvärliga simuleringar. Min redovisning hade kunnat göras väsentligt mer avancerad men min målsättning var bara att teckna en idé, ett användningsområde för hobbydatorn VIC-20. Jag hoppas att jag stimulerat någon eller några andra att ta sin VIC-20 till hjälp i forskningen, både på amatör- och yrkesnivå. På amatörnivå därför att många VIC-20 ägare förmodligen har en gryende forskare inom sig. Jag tror nämligen att intresse för datorer och forskning kan hänga intimt samman. Låt mig snart få läsa om Era resultat i VIC-Rapport.

Bernt Ekström
Lunds tekniska högskola

GRATISANNONSER

VIC 64 * PROGRAM * BYTES
Jag önskar byta program, har cirka 50 program sänd en lista till Pär-Uno Nilsson, Pl 5281, 441 90 Alingsås.

BYTES, SÄLJES VIC-20
Spel. Jag har bl a Dracula 16 K, Krazy Kong 16 K, Wachy Waiters och Grafik spel, och mycket annat. Skriv el ring till Tommy Lindroos, Vasamatie 19, 334 50 Siivikkala, Tammerfors, Finland. Tel. 009/358/31/46 09 40.

Vill Du ha list, print, load, run, save m m på funk-tangenterna till VIC 64. Programkassett 35:— betalas till postgiro 466 09 98-8. Conny Haglund.

SÄLJER ELLER BYTES
Program till VIC-20. Spelen är av bra kvalité. Hans Roine, Linnéavägen 4, 197 00 Bro. Tel. 0758/41 415. Säkrast efter kl. 16.

TRAV VIC-64 BIORYTM VIC-64
TRAV: Spela vinnare och plats välgjort realistiskt spel 9 KB utan listskydd
BIORYTM: Är du i form? Sätt in 59 kr på PG 456 05 17-7. Ronny Larsson, 0383-158 96.

BEG. VIC-64 KÖPES
Ring 08/87 39 31

TILL VIC-64
Ultrabasic, 50 st extra ljud och grafikkommando + Superskramble för endast 200:—. Nypris 420:—. Tel. 040-96 99 90.

TILLBEHÖR TILL VIC-20
Expantionslåda 6 uttag 500:—. Expantionskort med inbyggt 8K minne, 2 uttag 400:—. Micro Assembler med monitor, Programmers aid och höggrafikkommandon. 500:— 8K minne 300:—, VIC-graf 250:—. Texas instruments 59 med Mathutility modul + Master Libmodul 800:—. Texas Instruments Programmer, räknar i Decimalt, Oktalt, Hexadecimalt 350:—. Miche 08/61 76 59, 026/12 26 02.

VIC-20
Säljes: Cartridge: Avanger, Poker, Jupiter lander 90:— st eller 225 för alla: ZAP! POW! BOOM! Arcade games for VIC 20 75:—. Programmeringshandboken: 75:—, Bygg ut din VIC: 50:—, Basic på VIC-20: 30:—, Niklas Björklund, Bråviksvägen 18, 603 65 Norrköping, 011-11 82 95.

SÄLJES
SEX st kassettband C-20 för dataprogram säljes för endast 45:— insatt på pg 85 99 92-0, Lars Erik Larsson, Box 14, 665 00 Kil. Önskas mot postförskott tillkommer 8:—.

VIC 20 — MARS LANDER
En ny version av detta kända spel. Du skall genom stora grottsystem och landa rätt. Kämpa mot bränslebrist, eldklot, naturen, hinder m m. Flera olika kartor. Ordp. 95 kr, Nu: 55 kr. Lars Olsson, L. Anrås 6511, 444 00 Stenungsund.

VIC 64 SPEL BYTES
Scramble och Supergridder, båda köpta maskinkod pgm bytes mot liknande spel. Mikael Johansson, Cyklamenstigen 7, 432 00 Varberg.

KÖPES
Beg. VIC-64. 0381/112 12.

VIC-20 SÄLJES
Dator + bandspelare. ÅÄÖ. Superexpander. Plug-inspel: Jubiter Lander. VIC kassett 1&2. 4 böcker. En hel del program. Ring fr. efter 17 och helger till 0456-313 14. Pris: 3 000:—.

SPEL TILL VIC-20 BYTES
Ring till Robert Jansson efter kl. 17 00. Tel: 0121/138 18.

VIC-20 SPEL
3D Time Trek 16 K 95:—, Krazy Kong 16 K 95:—, Skyhawk 3 K el. 8 K 95:—, Flipper oexp. 80:—, Night Crawler oexp. 70:—, Race Fun oexp. 70:—. Mats 0150-513 15 efter 17.30.

****VIC-64 PROGRAM BYTES****
Häftiga spel i maskinkod och superba nyttoprogram bytes mot allt som kan köras på bandstation. Tel: 0171/-309 36 Petter eft. 16.

BYTES
VIC 64 program bytes. Jag skulle vilja byta program och erfarenheter. Tel. 0171-358 99 Jocke.

BREVVÄNNER SÖKES
Äger du en VIC-20 och vill ha brevvänner att byta erfarenheter och program??? Skriv då till Leif Lindström, Dahlanderg. 39, 703 48 Örebro.

VIC-64 SPEL BYTES
Fyra spel (värde 450 kr) Frogger, Panic, Scramble, Gridrunner bytes mot Cartridge spelet Cupfinal. Var du än bor ring 08/776 18 87. Fråga efter Charlie.

SPEL-CARTRIDGE
till VIC-20. Super Slot. Suveränt ljud och grafik. Säljes t. förmånspriset 150:— Best. till Magnus Hyltén, Box 129, 232 02 Åkarp. Tel. 040-46 48 46.

VIC-20-HANDBOKEN
Erbjudande: Ny användarmanual säljes till förmånspriset 45:— OBS! Jättebilligt Best t. Magnus Hyltén, Box 129, 232 02 Åkarp. Tel. 040-46 48 46.

VILLIG VIC-20 SÄLJES
VIC-20 med svenska tecken, superexpander, maskinkodsmonitor, användarmanual, programmering handbok och VIC-revealed. Massor av spel bl a Skramble, Exterminator, Snakman och Gridrunner på kassett. Jelly monsters och Rat Race på pluginkassett. Joystick medföljer. Nypris c:a 5 000 kr. Nu endast 2 995 kr. Skriv snarast till Pär Bergström, Alabamagatan 2, 593 00 Västervik eller ring 0490/112 67 efter kl. 16.

SV. MANUAL TILL 64:AN
Svensk användarmanual till VIC-64 50:—. VIC-news år 1 nr 1, 2, 3, år 2 nr 1, 10 :—/styck. Tel 013-17 02 32.

PROBLEM MED DIN MPS-801?
MPS-801 kan krångla lite mot VIC-20 ett maskinprogram från mig löser problemet information mot Porto. Ulf Larsson, Söndrumsv. 60, 302 39 Halmstad.

KÖPES
VIC BANDSPELARE, max: 200:—300:—. Andreas Elner, ErikDahlbergsgatan 8, 211 48 Malmö.

VIC-20/64 RESETKNAPP
Ingen lödning eller annat ingrepp. Fungerar ej tillsamans med printer eller flexskiva. Pris 25:— på pg 478 95 08-1, eller skicka kontant till Peter Månsson, Fårhagsvägen 18, 583 21 Linköping.

VIC-20 TILLBEHÖR
VIC-1110 8 K Ram expansionsminne 350:—, VIC-1922 Cosmic Druncher spelmodul 250:—. Tel. 026/25 67 69, Göran Sandström.

VIC-20
Bonzo, Amok, Millipede, Roadtoad + läroboken från spel till Basic — kassett med 7 st. spel. Allt detta säljes för endast 300:—. Tel: 0591-136 76.

SÄLJES: VIC 20
VIC 20 med bandspelare, Superexp., böcker och kassetter med program. Pris: 2 500 kr. 0920-547 46. Lars-Göran.

VIC 20 TILLBEHÖR
Basic, Grafik och ljud på VIC 20 45:— st, VIC-Slot plugin: 100:7, Kemiprogram 8 K 75 kr. Glosprogram 8 K 75 kr. Ring gärna för mer info. på tel. 08-768 44 21 eller skriv till: Andreas Jakibik, Sätraängsväg 136, 182 37 Danderyd.

SÄLJES VIC-64
VIC-64 med 14'' Dux färgTV säljes för 5 000:— (nypris 6 595:—), använd 3 mån. VIC-64an utan TV 3 300:—. 10 program på kassett medföljer. Tel. 026-791 20.

VIC-20 SPEL BYTES
VIC 20 spel bytes. Stort sortiment. Skicka lista över dina spel till: Karl Pettersson, Avstyckningsvägen 24, 175 43 Järfälla. Obs. skriv telnr.

VIC-64 BANDIT-SPEL
Säljes ett nytt program på kassett + frakt. Pris 80 kr. S. Muto, Sernandersvägen 15—311, 752 62 Uppsala.

VIC-20 + TILLBEHÖR
VIC-20 + 3, 8, 16 K-minne + Reset + 5 st handböcker + 40 st maskinkodsspel + nyttoprog och div. Basic prog. + många goda råd och ett glatt humör. Nypris 4.600:— Nu endast 3 200:—. Jocke Cederblad, 0758/-171 96.

FÖR RADIOAMATÖRER
RTTY-Program till VIC-64. Programmet arbetar med Split-Screen, skriv ditt svar medan du tar emot. QTH-lokatorprogram, nya och gamla systemet. Tomas, tel. 0515/165 30 e. 18.00.

TILLFÄLLE! VIC 20 SPEL
Jelly Monsters + andra toppspel på plugginkassett säljes eller ev bytes mot andra spel. Tel. 0980-818 75.

DUBBELFLOPPY*VIC-64
Floppy 3040 samt interface ''Interpod'' (se ex annons i Compute nr 9/83 sid 129). Interpod fungerar till alla CBM-Floppys, printers, m m. Pris: Floppy ca 4 000:— samt för interpod ca 1 000:— Ring Åke Setréus, 08-39 26 86 kv.

****16-K KÖPES****
16-K minne till VIC-20 köpes för ungefär 400 kr tel. 08-768 24 66.

KÖPES
Gamla nr av Compute! och Compute!s Gazette köpes. Urban Svanberg Karlslundsgatan 60, 151 60 Södertälje. Tel. 0755-840 33 eller 655 96.

VIC-20 TILLBEHÖR SÄLJES
VIC 1515 printer med papper 2 095:—, VIC 1540 Floppy disk 2 495:—, Super expander 325:—, VIC 20 programmeringshandbok 85:—. Ring Johan Ekener, tel. 035/11 33 10 på kvällstid.

VIC 20 TILLBEHÖR
Expansionsenhet VIC 1010 700:— 40/80 tecken kretskort 500:—, Makinspråksmonitor VIC 1213 200:—, bok Assebler på VIC 20 50:—. Per Ekstig, tel 026-10 37 97.

SÄLJES VIC-20 SPEL
En kassett med 5 spel Smack-man, Skramble, Cosmlads, Snakeybuilding, dessutom ett Glosövningsprogram. Pris 50 kr. Skicka pengarna till Pablo Grate, Rosenbergsgatan 8, 185 00 Waxholm, så kommer kassett.

VIC 1515 PRINTER
Använd men perfekt. 1 250:—. Tel. 031-29 66 18 eft. 17 (V. Frölunda)

VIC-64 PROGRAM
Spel och andra program till VIC-64 köpes! Skicka en lista med priser och vad du har för program till: Joachim Allermalm, Vallåkravägen 128, 261 90 Landskrona.

VIC 20
Bra adventurespel i maskinkod köpes ej textspel. Ring 031-29 38 72, fråga efter Johan.

ADVENTURE SÄLJES
Mission Impossible-cartridge 150 kr. Lösning till ovanst. 10 kr. Ledtrådar 5 kr. Tel. 013/15 1017 eft. 17.00

VIC-20 SPEL BYTES
För 3, 8, 16 K. (gärna grafik adv.) Tel. 0753-257 79, Anders.

VIC 20 ASSEBLER
Plugg-in kassett inkl. 11-sidig manual. 20 nya kommandon, Labels. 349:—, frakt tillkommer. Skriv för ytterligare info. till Olof Johansson, Scheelegatan 18, 521 00 Falköping.

SÄLJES
VIC 20, manual, bandspelare, Basic på VIC, progr handbok, introduktion till basic 1 samt 6 band, spel o nytto + 1 st joystick. Slumpas för 1.600:— + frakt. Kenneth Adelstål, Bergåsvägen 38, 224 00 Anderstorp, tel. 0371/158 08.

VIC-20 spel säljes 3 st för 50 kr. Laddes + 8K, Lotto med rättning, Dodgecars, Boss Puzzle, Robot Attack, Skramble, Maximum + 3K, Amok (JS), Defender + 8K, Meteor Blaster, Mini Pacman. Vilka vill du ha? Pg 459 26 40-9 Tore Gullstrand.

VIC-20 VIC-64
VIC-20: Cartridge Star Battle 150:—, eller byte mot VIC-64 Cartridge, har även kassettspel till VIC-20, VIC-64. Program bytes ev. säljes. Kontakta Erik Bertell, Ringv. 23, 824 00 Hudiksvall, tel. 0650-161 74.

VIC 64 S

En glad nyhet!

VIC 64 S(vensk)

Vic 64 i svensk version har Å-Ä-Ö-tecken inbyggda redan vid köpet, dessutom ingår användarmanual på svenska i priset.

Av Vic 64 i svensk version fordras godkända anslutningsenheter. Transformator godkänd av SEMKO och video-dator-omkoppare (Televerkets krav) medföljer också (se bild).
Av datorer som säljs i Sverige krävs att de inte överskrider de hårda svenska normerna för radiostörstrålning. Vic 64 är avstörd och godkänd enligt Televerkets normer!









3.995:-

Cirkapris inkl. moms för centralenhet.

handic electronic ab

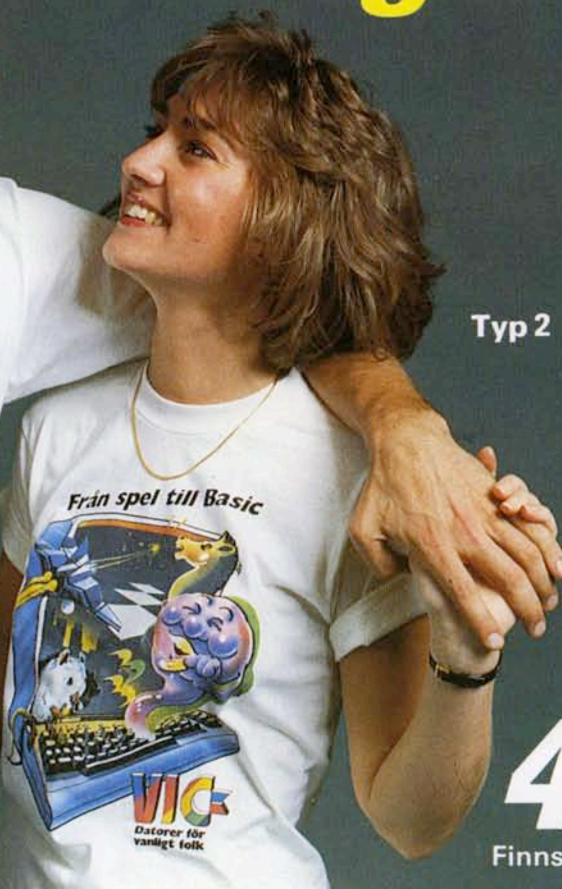
Box 1063, 436 00 Askim/Göteborg, Tel. 031/28 97 90
– ett företag i Datatronicgruppen –

tröjor för VIC ägare

Typ 1



Typ 2



Beställ med prenumerationskupongen

49:-

Finns i alla storlekar.

WICO. Världens bästa joysticks och manöverkontroller för datorer.

Wico passar till Commodore Vic 20, Vic 64, Apple, Atari, Coleco, Mattel- Intellivision, Texas Instruments, TRS -80, IBM PC m.fl.



WICO
THE SOURCE

MARKNADSFÖRS I NORDEN AV DENNIS BERGSTRÖM TRADING AB, TORSTENSSONSGATAN 4, BOX 14204, 104 40 STOCKHOLM. TELEFON 08-67 96 35. TLX 105 67 DEBE S.